

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04N 7/30
H04N 1/41

[12] 发明专利申请公开说明书

[21] 申请号 99808012.8

[43] 公开日 2001 年 8 月 8 日

[11] 公开号 CN 1307782A

[22] 申请日 1999.6.3 [21] 申请号 99808012.8
[30] 优先权
[32] 1998.7.3 [33] JP [31] 189239/1998
[86] 国际申请 PCT/JP99/02962 1999.6.3
[87] 国际公布 WO00/02393 日 2000.1.13
[85] 进入国家阶段日期 2000.12.28
[71] 申请人 株式会社哈德森
地址 日本北海道札幌市
[72] 发明人 三浦高志 板垣史彦 川岛深雪

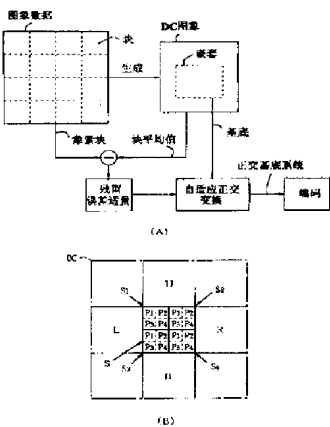
[74] 专利代理机构 中原信达知识产权代理有限责任公司
代理人 方挺余 滕

权利要求书 2 页 说明书 33 页 附图页数 17 页

[54] 发明名称 图象编码/解码方法以及记录该程序的记录媒体

[57] 摘要

本发明涉及图象编码/解码方法以及记录其程序的记录媒体,保持图象数据的高图象质量并且提高数据压缩率,同时能够减轻解码时的计算负荷。在图 1(A)中,分割图象数据来生成由各块平均值组成的 DC 图象,从各像素块分离出对应的块平均值,来求出每块的残差矢量,当残差矢量的幅值为允许值以上时,通过使用 DC 图象的嵌套的适当正交变换,来求出用于近似残差矢量的正交基底,进行编码。在图 1(B)中,当残差矢量小于允许值时,取代对正交基底进行编码,在解码侧,根据包含由 4×4 像素的块平均值组成的块 S 的上下左右 U,B,L,R 的共计 5 组 DC 图象数据,通过不分段的交流成分预测法来直接再现块 S 上的 4×4 像素的各像素数据。



权 利 要 求 书

1. 一种图象编码方法，其特征在于，把图象数据分割成多个象
素块，而生成由各个块的平均值组成的 DC 图象，从各个象素块分离
出对应的块平均值，来求出每个块的残差矢量，同时，当残差矢量的
幅值大于或等于允许值时，通过使用 DC 图象的嵌套的自适应正交变
换，来求出用于近似残差矢量的一个或者两个以上的正交基底，对由
它们的线性组合组成的正交基底系统进行编码。

2. 根据权利要求 1 所述的图象编码方法，其特征在于，当残差
矢量的幅值小于允许值时，对基底数=0 的信息进行编码，以取代求出
正交基底的过程。

3. 根据权利要求 1 所述的图象编码方法，其特征在于，当求出
的正交基底系统的总代码量大于等于残差矢量的总代码量时，对残差
矢量进行编码，以取代正交基底系统的编码。

4. 根据权利要求 1 所述的图象编码方法，其特征在于，通过预
先求出的基底数 nk ，当与残差矢量 $\langle d \rangle$ 的第一误差 $\langle d_{nk} \rangle$ 成为小于
允许值时，把按上述所求出的基底顺序使用 m ($0 \leq m < nk$) 个基底之
后的第二误差矢量 $\langle d_m \rangle$ ，用根据未使用的其余基底数 yk ($=nk-m$)
而预先决定的量化系数 Q_{yk} 来进行标量量化，并且，把其进行标量逆
量化；从用上述第二误差矢量 $\langle d_m \rangle$ 减去标量逆量化的第二误差矢量
而得到的 nk 种类的第三误差矢量 $\langle d'_m \rangle$ 和上述第一误差 $\langle d_{nk} \rangle$ 之
中，选择最小的误差；同时将对应的基底以及必要时的第二误差矢量
来进行编码。

5. 一种图象编码/解码方法，其特征在于，根据包含由 $K \times K$ 象
素的块平均值组成的自身块 S 的上下左右 U,B,L,R 的共计 5 个 DC 图
象数据，按下式求出自身块 S 上的左上的第一小块 S_i 的 $(K/2) \times (K/2)$

的各像素数据 $P_1 \sim P_4$:

$$P_1 = S + (2U + 2L - 2S - B - R) / 8$$

$$P_2 = S + (2U - B - R) / 8$$

$$P_3 = S + (2L - B - R) / 8$$

$$5 \quad P_4 = S + (2S - B - R) / 8$$

并且,按下式求出自身块 S 上的右上的第二小块 S_2 的 $(K/2) \times (K/2)$ 的各像素数据 $P_1 \sim P_4$:

$$P_1 = S + (2U - B - L) / 8$$

$$P_2 = S + (2U + 2R - 2S - B - L) / 8$$

$$10 \quad P_3 = S + (2S - B - L) / 8$$

$$P_4 = S + (2R - B - L) / 8$$

并且,按下式求出自身块 S 上的左下的第三小块 S_3 的 $(K/2) \times (K/2)$ 的各像素数据 $P_1 \sim P_4$:

$$P_1 = S + (2L - U - R) / 8$$

$$15 \quad P_2 = S + (2S - U - R) / 8$$

$$P_3 = S + (2B + 2L - 2S - U - R) / 8$$

$$P_4 = S + (2B - U - R) / 8$$

并且,按下式求出自身块 S 上的右下的第四小块 S_4 的 $(K/2) \times (K/2)$ 的各像素数据 $P_1 \sim P_4$:

$$20 \quad P_1 = S + (2S - U - L) / 8$$

$$P_2 = S + (2R - U - L) / 8$$

$$P_3 = S + (2B - U - L) / 8$$

$$P_4 = S + (2B + 2R - 2S - U - L) / 8。$$

25 6. 一种能够进行计算机读取的记录媒体,记录用于由计算机执行权利要求 1 至 5 任一项所记载的处理的程序。

说明书

图象编码/解码方法以及记录该程序的记录媒体

5 技术领域

本发明涉及图象编码/解码方法以及记录该程序的记录媒体，更详细地说，涉及能够用游戏机等性能较低的硬件（CPU、存储器等）来高速再现高质量的图象（CG 图象、动画片图象、自然图象等）的图象编码/解码方法以及记录该程序的记录媒体。

10 背景技术

图 16、图 17 是说明现有技术的图（1）、（2），图 16 表示由 JPEG 所产生的图象压缩处理的图象。

15 在图 16（A）中，在目前成为静止图象压缩方式的主流 JPEG 中，把图象分割成 8×8 像素的块，把其通过二维 DCT（离散余弦变换）变换为 DC（平均值）和从基本频率至 63 倍频率的各个系数值。而且，利用自然图象的频率成分集中在低频区域的特点，在画面质量不会降低的范围内，用不同的量化幅度对各系数值进行量化，以削减信息量，然后进行可变长度编码（霍夫曼编码）。

20 另一方面，当在家用的游戏机等中利用这样的图象编码数据时，由于受 CPU 性能和存储器容量上的限制，当用软件对解码运算负荷较大的图象压缩法（JPEG 等）进行解码时，会产生各种缺点。例如，25 在 JPEG 中，对于 8×8 像素的块，必须生成应进行 64 个可变长度编码的代码，因此，引起解码时的计算负荷增大。

图 16（B）表示霍夫曼代码的一部分。

30 在霍夫曼编码中，出现频率高的系数值 H_{13} 等以相对短的代码长

度比特进行编码，出现频率低的系数值 H_6 等以相对长的代码长度比特进行编码，因此，这些代码在各个八位（字节）上如图所示的那样封装得参差不齐，使解码时的计算负荷变得非常大。因此，在现有的游戏系统中，存在为了以作为活动图象能够容忍的程度的速度来复原图象，而不得不过于牺牲图象质量的情况。

而且，对于图象质量，频率成分越高，量化的精度就越低。因此轮廓部分的图象信息丢失，而发生蚊噪声，不适合于文字和动画图象的压缩。特别是，在游戏软件中，大多使用人工图象（CG 图象、动画图象等），因此，由蚊噪声所引起的主观的图象质量变差成为较大的问题。

对此，近年来，报告了以下文献（1）～（5）：

（1）Michael F. Barnsley, Lyman P.Hurd “FRACTAL IMAGE COMPRESSION（分图像压缩）” AK Peters Ltd.,1993

（2）Takashi Ida, Takeshi Datake “Image compression by Iterative Transformation Coding（通过迭代转换编码进行图像压缩）”，《第五届电路和系统 Karuizawa 工作站论文集》，第.137-142 页,1992

（3）Hyunbea You, Takashi Takahashi, Hiroyuki Kono, Ryuji Tokunaga “Improving LIFS Image Coding Scheme by Applying Gram Schmidt Orthogonalization(通过采用 Gram Schmidt 正交方法改进 LIFS 图像编码方案)”，《电子、信息和通信学会技术研究報告》，第 NLP-98 卷，第 146 期，第 37-42 页,1998 年

（4）Toshiaki Watanabe, Kazuo Ozeki “Study of AC Component Prediction Scheme Using Mean Values（采用平均值的交流成分预测方式的研讨）” 图像编码研讨会（PCSJ89），第 29-30,1989 年 10 月；和

（5）Takashi Takahashi, Ryuji Tokunaga “High-Speed Computation Algorighm for predicting AC Components from Block Mean Values of Images（用于从图像的块平均值预测交流成分的高速计算算法）” 电

子信息通信工程学会论文集,第 J81-D-II 卷,第 4 期,第 778-780 页 1998 年 4 月。

其中,方案 (1), (2) 涉及作为解码运算负荷较小的压缩方式的分形 (fractal) 编码,方案 (3) 涉及具有与 JPEG 相同程度的编码效率的自适应正交变换的改进,而且,方案 (4), (5) 涉及根据块平均值 (DC 值) 的交流成分预测。

其中,方案 (3) 是这样的块编码法:把图象分割成 $K \times K$ 像素的正方形块,根据允许误差 Z ,用交流成分预测法、分形变换法、自适应正交变换中的任一种来近似全部的块。交流成分预测法被用于平均值分离方式:从包围自身 (local) 块的周围块的平均值 (DC 值) 来求出自身块的交流成分 (相加数据),对其与对象图象的残差进行编码。而且,自适应正交变换是这样的方式:利用图象具有的自相似性,从相当于矢量量化的代码簿的图象 (嵌套) 抽出用于近似块图象的基底矢量,通过 GRAM SCHMIDT 法来构成必要的最小维数的正交基底系统。

但是,上述方式 (1), (2) 的分形变换,在解码中需要进行反复计算,并且,需要消耗图象平面规模的工作空间,因此,不利于视频游戏机等。

而且,在方式 (3) 的自适应正交变换中,利用与对象图象同等大小的加法运算数据作为嵌套,在解码时,需要非常大的工作空间。而且,在解码时,把被解冻的块依次地写回到嵌套上的对应块上,因此,使画面质量得到改善,但是,地址计算和数据传送的负荷加重。而且,在方式 (3) 中,在基底的坐标和取样系数的压缩中使用霍夫曼代码,但是,在自然图象中,在何种基底的出现频率上,没有较大的偏向,因此,压缩效率不会提高,或者,消费了霍夫曼解码的计算量。而且,在自适应正交变换中,随图象会存在必要的最小维数的正交基底数变多的情况,则当基底数变多时,就需要使用比对残差矢量

进行直接编码时更多的比特数，则编码效率降低。

而且，在方式（4）的交流成分预测法中，存在在图象中的轮廓部分周边产生过冲和下冲的倾向，存在在亮度急剧增加的人工图象中，反而会使图象质量变差的倾向。

而且，在方式（5）的分段交流成分预测方式中，不仅 CPU 负担加重，而且需要中间生成的插补值的存储区域。

在图 17 中表示分段交流成分预测方式的图象。

在图 17（A）中，在分段交流成分预测方式中，通过下式从包含注目块的周围 4 块的各 DC 值（S,U,R,B,L）推断所注目块 S 上的小块 $S_1 \sim S_4$ 的各 DC 值：

$$S_1 = S + (U + L - B - R) / 8$$

$$S_2 = S + (U + R - B - L) / 8$$

$$S_3 = S + (B + L - U - R) / 8$$

$$S_4 = S + (B + R - U - L) / 8$$

在图 17（B）中，通过重复使用上述方式，能够进一步通过下式推断出小块 S_1 上的四个象素 $P_1 \sim P_4$ 的各象素值：

$$P_1 = S_1 + (U_3 + L_2 - S_3 - S_2) / 8$$

$$P_2 = S_1 + (U_3 + S_2 - S_3 - L_2) / 8$$

$$P_3 = S_1 + (S_3 + L_2 - U_3 - S_2) / 8$$

$$P_4 = S_1 + (S_3 + S_2 - U_3 - L_2) / 8$$

同样，通过下式能够推断出小块 S_2 上的 4 象素 $P_1 \sim P_4$ 的各象素值：

$$P_1 = S_2 + (U_4 + S_1 - S_4 - R_1) / 8$$

$$P_2 = S_2 + (U_4 + R_1 - S_4 - S_1) / 8$$

$$P_3 = S_2 + (S_4 + S_1 - U_4 - R_1) / 8$$

$$P_4 = S_2 + (S_4 + R_1 - U_4 - S_1) / 8$$

同样，能够推断出小块 S_3 、 S_4 上的各 4 象素 $P_1 \sim P_4$ 的各象素值：

5

但是，为了从最初的 DC 值 (S, U, R, B, L) 出发，得到原图象的全部预测值 $P_1 \sim P_4$ ，不但需要分段地使用上述方式而不进行细分，不仅需要保持中间值的存储器，而且，CPU 的运算负荷增大。

10

发明概述

鉴于上述现有技术的问题，本发明的目的是提供图象编码/解码方法及记录该程序的记录媒体，对于可以在计算机上使用的图象数据，在保持较高的图象质量的同时，提高图象数据的压缩率，而且，能够减轻解码时的计算负荷。

15

通过例如图 1 (A) 的构成来解决上述问题。即，本发明方案 (1) 的图象编码方法，把图象数据分割成多个象素的块，而生成由各个块的平均值组成的 DC 图象，从各个象素块分离出对应的块平均值，来求出每个块的残差矢量，同时，当残差矢量的幅值为允许值以上时，
20 通过使用 DC 图象的嵌套的自适应正交变换，来求出用于近似残差矢量的一个或者两个以上的正交基底，对由它们的线性组合组成的正交基底系统进行编码。

20

在本发明 (1) 中，通过从 DC 图象的嵌套求出一个或者两个以上的正交基底的构成，而大幅度减轻了图象编码/解码时的 CPU 和存储器的负荷。而且，平坦的亮度成分较多，则在与 DC 图象的相关较强的 CG 图象和动画图象等人工图象中，可以期望提高图象质量。

25

优选地，在本发明方案 (2) 中，在上述本发明 (1) 中，当残差矢量的幅值小于允许值时，对基底数=0 的信息进行编码，以取代求出

30

正交基底的过程。

例如，一般对于亮度急剧增加的人工图象，对于不存在于轮廓部分附近的（即，残差矢量的幅值小于允许值的）象素块，有限制地生成基底数=0 的代码。这样，对于这样的象素块，能够得到高的图象压缩率。

另一方面，在解码侧，对于对应的象素块，通过有限制地使用分段的交流成分预测法或者本发明的不分段的交流成分预测法，能够以较少的 CPU 和存储器量来再现象素块。

优选地，在本发明方案（3）中，在上述本发明（1）中，当求出的正交基底系统的总代码量成为残差矢量的总代码量以上时，对残差矢量进行编码，以取代正交基底系统的编码。这样，不会无意中降低图象压缩率。

优选地，在本发明方案（4）中，在上述本发明（1）中，通过预先求出的基底数 nk ，当与残差矢量 $\langle d \rangle$ 的第一误差 $\langle d_{nk} \rangle$ 小于允许值时，把按上述所求出的顺序使用 m ($0 \leq m < nk$) 个基底之后的第二误差矢量 $\langle d_m \rangle$ ，用对应于未使用的其余基底数 $y_k (=nk-m)$ 所预先决定的量化系数 Q_{y_k} 来进行标量量化，并且，把其进行标量逆量化，从用上述第二误差矢量 $\langle d_m \rangle$ 减去标量逆量化后的第二误差矢量而得到的 nk 种类的第三误差矢量 $\langle d'_m \rangle$ 和上述第一误差 $\langle d_{nk} \rangle$ 之中，选择最小的误差，同时将对应的基底和成为必要的第二误差矢量合并使用来进行编码。

下面参照图 12 (A) 的 (c)，(d) 来具体说明一个例子的编码方法。在图 12 (A) 的 (c) 中，现在，求出使与残差矢量 $\langle d \rangle$ 的误差的幅值 $\| \langle d_{nk} \rangle \|$ 成为小于允许值 Z 这样的由 nk (例如 $nk=3$) 个基底组成的线性结合：

$$\langle d \rangle \approx \beta_1 \langle v_1 \rangle + \beta_2 \langle v_2 \rangle + \beta_3 \langle v_3 \rangle \quad (nk=3)$$

其中， $\langle v \rangle$ 表示 v 是矢量。

5 对于使用上述 $nk=3$ 个基底时的第一误差矢量 $\langle d_1 \rangle = \langle d_{nk} \rangle$ 和以
上述求出的顺序使用 $m(0 \leq m < nk)$ 个基底之后的第二误差矢量 $\langle d_0 \rangle \sim$
 $\langle d_2 \rangle$ ，具有下列关系：

$$\begin{aligned} \langle d_0 \rangle &= \langle d \rangle & (m=0, yk=3) \\ \langle d_1 \rangle &= \langle d \rangle - \beta_1 \langle v_1 \rangle & (m=1, yk=2) \\ \langle d_2 \rangle &= \langle d \rangle - \beta_1 \langle v_1 \rangle - \beta_2 \langle v_2 \rangle & (m=2, yk=1) \\ \langle d_3 \rangle &= \langle d \rangle - \beta_1 \langle v_1 \rangle - \beta_2 \langle v_2 \rangle - \beta_3 \langle v_3 \rangle & (m=3, yk=0) \end{aligned}$$

15 在图 12 (A) 的 (d) 中，通过对应于未使用的其余基底数 $yk=3,$
2, 1 所预先决定的量化系数 Q_{yk} (例如, $Q_3=6, Q_2=7, Q_1=8$)，分别
对上述第二误差矢量 $\langle d_0 \rangle \sim \langle d_2 \rangle$ 进行标量量化 (并进行剪切)，
并且，使用相同的量化系数 Q_{yk} 分别对它们进行逆量化，来求出逆量
化后的第二误差矢量 $\langle d_0 Q' \rangle \sim \langle d_2 Q' \rangle$ 。

$$\begin{aligned} \langle d_0 Q' \rangle &= [\langle d_0 \rangle / Q_3] \times Q_3 \\ \langle d_1 Q' \rangle &= [\langle d_1 \rangle / Q_2] \times Q_2 \\ \langle d_2 Q' \rangle &= [\langle d_2 \rangle / Q_1] \times Q_1 \end{aligned}$$

其中，记号 $[\]$ 表示将运算结果取整数。

25 接着，求出从上述第二误差矢量 $\langle d_0 \rangle, \langle d_1 \rangle, \langle d_2 \rangle$ 分别减
去它们而得到的第三误差矢量 $\langle d'_0 \rangle \sim \langle d'_2 \rangle$ 。

$$\begin{aligned} \langle d'_0 \rangle &= \langle d_0 \rangle - \langle d_0 Q' \rangle \\ \langle d'_1 \rangle &= \langle d_1 \rangle - \langle d_1 Q' \rangle \\ \langle d'_2 \rangle &= \langle d_2 \rangle - \langle d_2 Q' \rangle \end{aligned}$$

但是，这些第三误差矢量 $\langle d'_0 \rangle \sim \langle d'_2 \rangle$ 的幅值并不一定大于上述第一误差矢量 $\langle d_3 \rangle$ 。即，对三个基底 $\beta_1 \langle v_1 \rangle + \beta_2 \langle v_2 \rangle + \beta_3 \langle v_3 \rangle$ 进行编码，其结果，该解码后的误差成为上述第一误差
 5 矢量 $= \langle d_3 \rangle$ ，与之相比，例如，对两个基底 $\beta_1 \langle v_1 \rangle + \beta_2 \langle v_2 \rangle$ 进行编码，并且，利用其剩余的一个基底的代码量来对第二误差矢量 $\langle d_2 \rangle$ 进行标量量化，其结果，该解码后的误差成为上述第三误差矢量 $= \langle d'_2 \rangle$ ，采用这样的方法，最终的解码误差可能较小。

10 因此，在本发明方案（4）中，在第三误差矢量 $\langle d'_0 \rangle$ ， $\langle d'_1 \rangle$ ， $\langle d'_2 \rangle$ 与第一误差矢量 $\langle d_3 \rangle$ 之间，选择误差最小的那个，来对对应的基底和必要时的第二误差矢量进行编码。

在上述例子中，在第三误差矢量 $\langle d'_2 \rangle$ 的误差是最小的情况下，
 15 采用 $\{\beta_1 \langle v_1 \rangle + \beta_2 \langle v_2 \rangle\}$ 作为正交基底系统，把它们进行编码，同时，并行地对第二误差矢量 $\langle d_2 \rangle$ 来进行编码（标量量化）。确定量化系数 Q_{yk} ，以使这样生成的代码不会大于使用 nk 个基底时的总代码量，因此，不会使每个象素块的代码量增加，但又能够提高图象质量。

20 这样，在本发明（4）中，通过同时使用自适应正交变换和在使用必要时的 m 个基底后的第二误差矢量的标量量化的构成，就能以更高精度对残差矢量 $\langle d \rangle$ 进行编码。而且，此时，用与其余基底数 yk ($=nk-m$) 相对应的量化系数 Q_{yk} 来对第二误差矢量进行标量量化，
 25 并且，把其编码为与其余基底数 yk 相同的代码量，因此，不会使与该象素块相对应的总代码量增加，但又提高了图象质量，而且，能够使每个象素块的代码长度成为容易解码的形态（预定比特数的倍数），这样，能够大幅度减轻解码时的运算负荷。

30 由图 1（B）的构成所产生的本发明方案（5）的图象编码/解码方

法，根据包含由 $K \times K$ 象素的块平均值组成的自身块 S 的上下左右 U, B, L, R 的共计 5 个 DC 图象数据，按下式求出自身块 S 上的左上的第一小块 S_1 的 $(K/2) \times (K/2)$ 的各象素数据 $P_1 \sim P_4$ ：

$$P_1 = S + (2U + 2L - 2S - B - R) / 8$$

$$5 \quad P_2 = S + (2U - B - R) / 8$$

$$P_3 = S + (2L - B - R) / 8$$

$$P_4 = S + (2S - B - R) / 8$$

并且，按下式求出自身块 S 上的右上的第二小块 S_2 的 $(K/2) \times (K/2)$ 的各象素数据 $P_1 \sim P_4$ ：

$$10 \quad P_1 = S + (2U - B - L) / 8$$

$$P_2 = S + (2U + 2R - 2S - B - L) / 8$$

$$P_3 = S + (2S - B - L) / 8$$

$$P_4 = S + (2R - B - L) / 8$$

并且，按下式求出自身块 S 上的左下的第三小块 S_3 的 $(K/2) \times (K/2)$ 的各象素数据 $P_1 \sim P_4$ ：

$$15 \quad P_1 = S + (2L - U - R) / 8$$

$$P_2 = S + (2S - U - R) / 8$$

$$P_3 = S + (2B + 2L - 2S - U - R) / 8$$

$$P_4 = S + (2B - U - R) / 8$$

20 并且，按下式求出自身块 S 上的右下的第四小块 S_4 的 $(K/2) \times (K/2)$ 的各象素数据 $P_1 \sim P_4$ ：

$$P_1 = S + (2S - U - L) / 8$$

$$P_2 = S + (2R - U - L) / 8$$

$$P_3 = S + (2B - U - L) / 8$$

$$25 \quad P_4 = S + (2B + 2R - 2S - U - L) / 8$$

在本发明(5)中，通过从包含自身块的附近的 DC 图象数据 S, U, B, L 不分段地直接求出自身块 S 上的 $K \times K$ 象素的象素数据的构成，能够大幅度减轻图象编码/解码时的 CPU 和存储器负荷。而且，本发明(5)的方法，在图象编码时，可以用于交流成分的预测等，而在图象解码

时，能够用于交流成分的再现等。

本发明（6）的记录媒体，是能够进行计算机读取的记录媒体，记录用于由计算机执行上述本发明（1）～（5）任一项所记载的处理的程序。

附图的简要说明

图1是说明本发明的原理的图；

图2是表示根据实施例的图象编码/解码方式的构成图；

图3是表示根据实施例的图象编码处理的流程图（1）；

图4是表示根据实施例的图象编码处理的流程图（2）；

图5是表示根据实施例的图象编码处理的流程图（3）；

图6是表示根据实施例的图象编码处理的流程图（4）；

图7是表示根据实施例的图象编码处理的流程图（5）；

图8是说明由实施例所产生的图象编码处理的表的图；

图9是表示根据实施例的图象编码处理的图象（1）；

图10是表示根据实施例的图象编码处理的图象（2）；

图11是表示根据实施例的图象编码处理的图象（3）；

图12是表示根据实施例的图象编码处理的图象（4）；

图13是表示根据实施例的图象编码处理的流程图；

图14是表示根据实施例的不分段的交流成分预测处理的图象；

图15是根据实施例的编码效率的曲线图；

图16是说明现有技术的图（1）；

图17是说明现有技术的图（2）。

用于实施发明的最佳方案

下面按照附图来详细说明本发明的最佳实施例。并且，在全部附图中，相同的标号表示相同或者相当的部分。

图 2 是表示根据实施例的图象编码/解码方式的构成图，表示适



用于游戏系统的例子。

在图 2 中，10 是游戏机本体；11 是 CPU（游戏处理器），进行
游戏机的主控制（游戏程序、3D CG 和纹理数据 TXD 的读入控制、
由用户所进行的游戏操作数据的读入控制等）以及与 3D CG 动画相关的
游戏处理（游戏字符的移动处理等）；12 是由 CPU 11 使用的 RAM、
ROM、EEPROM 等组成的主存储器（MM）；13 是 CPU 11 的本机（主
机）总线；14 是能够高速传送数据的总线；15 是进行总线间的连接
（以及竞争裁决控制）的桥路；16 是操作面板，包括用于用户进行游
戏操作和参数设定的各种控制键（开始键、选择键、十字键等）以及
鼠标器、游戏棒等位置输入装置；17 是把操作面板 16 容纳在游戏系
统中的周边接口（PIF）；18 是把游戏程序（处理程序、游戏参数、3D
CG 模型数据、纹理数据等）记录在屏蔽 ROM 等中的 ROM 卡盒（ROM
-C）；19 是能够把 ROM 卡盒 18 装卸自如地装入游戏系统中的 ROM
卡盒接口（ROM-CIF）；20 是记录上述游戏程序的光盘 ROM（CD
-ROM）；21 是能够把光盘 ROM 20 装卸自如地装入游戏系统中并
进行驱动的 CD-ROM 驱动器；22 是把本系统连接到未图示的公众网
上的通信控制部（COM），用于进行游戏程序的在线下载和与其他游
戏机之间的对抗游戏；23 是通信线。

而且，30 是用于把输入的 3D CG 模型数据透视变换为 2D 画面
的 3D 加速器；31 是几何图形处理器；32 是模型变换部，根据字符、
背景等各基准位置数据及其形状模型数据，在 3D 空间上的各对应位
置上生成和展开由多边形等集合组成的实际尺寸的 3D CG 模型；33
是给上述生成的模型附加考虑光线效果（太阳光、照明光等）的阴影
的阴影处理部；34 是视点变换部，把世界坐标系 XYZ 中的 3D CG 图
象数据变换为符合摄象机视点信息的视点坐标系 xyz 的 3D CG 图象数
据；35 是透视变换部，根据摄象机的视点信息（视点坐标系 xyz）来
进行 3D CG 生成模型的透视变换。

36 是进行与显示光栅同步的处理的透视图（光栅）处理器；37 是跟据 CRT 45 的扫描线地址 x,y 而抽出透视变换后的各像素数据的扫描线分解部；38 是 Z 缓冲器；39 是隐藏面消去部，使用 Z 缓冲器 38，用在画面 45A 上所见到部分的像素数据替换在画面 45A 上不应当看到的部分的像素数据；40 是纹理解码部，对本发明的输入的纹理压缩数据进行解冻（解码）；41 是暂时存储纹理数据的纹理缓冲器；42 是给在画面 45A 上见到的部分的各像素数据粘贴纹理缓冲器 41 的各个对应的纹理数据的纹理处理部；43 是存储输出的彩色图象数据（R,G,B）的帧存储器；44 是 D/A 变换部（DAC），把帧存储器 43 的彩色像素数据变换为模拟像素信号，并给 CRT 45 提供视频信号 VS；45 是作为显示部的例如彩色 CRT 监视器；45A 是其显示画面；46 是发生 CRT 45 的水平垂直同步信号 H,V 以及 3D 加速器 30 的 x,y 同步信号的 CRT 控制部（CRTC）。

外部的游戏制作装置 50 制作游戏程序（处理程序、游戏参数、3D CG 模型数据、纹理数据等），并记录在 ROM 卡盒 18 或者 CD-ROM 20 中。而且，在制作纹理数据时，执行本发明的图象编码方式，进行了图象压缩的纹理数据 TXD 被记录在 ROM 卡盒 18 或者 CD-ROM 20 中。接着，该被图象压缩的纹理数据 TXD 由纹理解码部 40 进行高速解码，并且，被粘贴到 2D 透视图象上，在画面 45A 上进行显示。以下，说明游戏制作装置 50 中的图象编码（压缩）处理。

图 3～图 7 是根据一实施例的图象编码处理的流程图（1）～（5），图 8 是说明该图象编码处理的表的图，图 9～图 12 是该图象编码处理的图形（1）～（4），以下参照这些图来说明图象编码处理的详细情况。

在整个说明书中，记号 $\langle \rangle$ 表示矢量，记号 $\| \quad \|$ 表示矢量的幅值，记号 \cdot 表示矢量的内积，记号 [运算] 表示运算结果的整数化。其中，当在不同意义下使用记号 [运算] 时，附加其意思。此外，用加粗

的字符来表示图和等式中的矢量。

在流程图中，使原始图象的尺寸为横向 n 、纵向 m 个像素。而且，
 记号 $T_{m,n}$ 表示从图象 T 的 m 行 n 列取出 4×4 像素的正方形块的部分
 5 图象数据，记号 $B_{y,x,sy,sx}$ 表示从 DC 图象的 y 行， x 列在横向上以 sx 步
 幅、在纵向上以 sy 步幅取出 4×4 的共计 16 像素的数据的块。

图 3 表示图象编码处理的主处理。

10 在步骤 S1 中，读入原始图象数据。

在图 9 (A) 中表示原始图象数据 T 的图形。把 RGB 系的对象图
 象变换为 YUV 系来读入。Y 对应于亮度数据，U,V 对应于色差数据，
 使用横向 2 像素的亮度平均来对 U、V 进行降频取样 (downsample)。
 15 一个例子的亮度数据 Y 由纵 $960 \times$ 横 1280 像素组成，U、V 分别由纵
 $960 \times$ 横 640 像素组成。给 Y、U、V 的各个像素数据分别分配例如 8
 比特。

以下以亮度数据 Y 的处理为中心进行描述，但是，对于 U、V，
 20 可以进行同样的处理。

回到图 3，在步骤 S2 中，执行 DC 图象制作处理。DC 图象制作
 处理是这样的处理：把原始图象分割成每个 4×4 像素的块，制作由
 各块的平均值 M 组成的 DC 图象。

25 在图 4 中表示了 DC 图象制作处理的流程图。在步骤 S21 中，把
 原始图象的地址寄存器 i,j 同时初始化为「0」。在步骤 S22 中，把 DC
 图象的地址寄存器 I, J 中设置为 $i/4, j/4$ 。在步骤 S23 中，从原始图
 象取出 4×4 像素的块数据 $T_{j,i}$ 。在步骤 S24 中，求出在块数据 $T_{j,i}$ 中
 30 包含的 16 像素数据的平均值 M 。在步骤 S25 中，把该平均值 M 存储

到 DC 图象的存储位置 $DC_{j,i}$ 中。在步骤 S26 中, 执行 $i+4$, 在步骤 S27 中, 判别是否 $i>n$ 。当不是 $i>n$ 时, 返回步骤 S22, 这次求出下一列的块数据 $T_{j,i}$ 的平均值 M , 存储到下一列的 $DC_{j,i}$ 中。以此类推, 即, 在步骤 S27 的判别中, 当为 $i>n$ 时, 在步骤 S28 中, 把 i 初始化为「0」, 且执行 $j4$ 。在步骤 S29 中, 判别是否 $j>m$, 当不是 $j>m$ 时, 返回步骤 S22, 这次求出下一行的块数据 $T_{j,i}$ 的平均值 M , 存储到下一行的 $DC_{j,i}$ 中。以此类推, 即, 在步骤 S29 的判别中, 当为 $j>m$ 时, 退出该处理。

在图 9 (B) 中表示 DC 图象数据 DC 的图形。一个例子的 DC 图象由纵 $240 \times$ 横 320 的 DC 值构成。

返回图 3, 在步骤 S3 中, 通过二维 DPCM 对 DC 图象进行编码并输出。

在图 10 (A) 中表示二维 DPCM 的处理图形。当 DC 图象的 J 行, I 列的 DC 值为 $DC_{j,i}$ 时, 通过例如 $DC'_{j,i} = (DC_{j,i-1} + DC_{j-1,i}) / 2$ 来求出该 $DC_{j,i}$ 的预测值 $DC'_{j,i}$, 通过量化系数 Q_s 来对其预测误差 $\Delta DC_{j,i} = DC_{j,i} - DC'_{j,i}$ 进行标量量化 {即, $[\Delta DC_{j,i} / Q_s]$ }, 并输出。其中, 在此情况下的记号 $[a]$ 表示对实数 a 进行四舍五入的结果。而且, 仅在预测误差 $\Delta DC_{j,i}$ 为「0」的情况下, 考虑扫描宽度, 分别独立地对预测误差 $\Delta DC_{j,i}$ 和扫描宽度进行霍夫曼编码。

在图 8 (A) 中表示量化系数 Q_s 的表。量化系数 Q_s 的值与多段的允许误差 Z 相对应。使用者在要求高画面质量时允许误差 Z 范围选择得小些, 在较低的画面质量也可以时, 允许误差 Z 范围选择得大些。据此, 量化系数 Q_s 在 1~8 的范围内进行变化。

返回图 3, 在步骤 S4 中, 原始图象的地址寄存器 i, j 同时初始化为「0」。在步骤 S5 中, 把 DC 图象的地址寄存器 I, J 中设置为 $i/4$,

j/4。在步骤 S6 中，通过下式来求出与 4×4 像素的像素块相对应的 16 维的残差矢量 $\langle d \rangle$ ：

$$\langle d \rangle = T_{j,i} - DC_{j,i}$$

5 在图 10 (B) 中表示残差矢量 $\langle d \rangle$ 的图形。给残差矢量 $\langle d \rangle$ 的各个部分分别分配 8 字节。

10 返回图 3，在步骤 S7 中，判别残差矢量 $\langle d \rangle$ 的幅值（平方）是否小于允许误差 Z 。在小于允许误差 Z 的情况下，在解码侧，能够通过后述的不分段的交流成分预测法来高精度地复原该部分的块图象数据 $T_{j,i}$ ，因此，流程进到步骤 S18，将基底数「0」作为代码 F_1 输出。在不小于允许误差 Z 的情况下，应当搜索用于近似残差矢量 $\langle d \rangle$ 的基底矢量，为此进到步骤 S8，执行自适应正交化处理。

15 在图 11 中表示了适当正交化处理的图形。适当正交化处理是这样的处理：为了在允许近似误差 Z 以内近似残差矢量 $\langle d \rangle$ ，而求出必要的基底矢量的个数 nk 和各基底矢量 $\langle v_{nk} \rangle$ 。在本实施例中，提取出以纵横 1: K （例如 $K=4$ ）压缩原始图象的 DC 图象的一部分，而作为嵌套使用，来谋求嵌套的轻量化。

20 在图 11 (A) 中，从 DC 图象切出例如纵 $39 \times$ 横 71 的 DC 值的区域，来作为嵌套。而且，基底矢量 $\langle v_{nk} \rangle$ 的搜索，在每个纵横 1DC 值上，设定顶点 $(x,y) \in [0, 63] \times [0, 31]$ ，并且，该二次取样间隔为 $(sx,sy) \in \{(1, 1), (1, 2), (2, 1), (2, 2)\}$ 共计 4 种。
25 而且，这是一个例子，嵌套的尺寸、切出位置、基底矢量的二次取样间隔等可以任意设定。

30 图 11 (B) 表示从嵌套以各种图形所取样的各 DC 值集中在基底矢量 $\langle v_{nk} \rangle$ 的 4×4 的存储区域中的状态。即，当 $(sx,sy) = (1, 1)$ 时，从嵌套上的 4×4 的区域，搜集 4×4 的 DC 值，并且，当 (sx,sy)

= (1, 2) 时, 从在嵌套上的 y 方向上延伸的区域, 搜集 4×4 的 DC 值, 并且, 当 $(sx, sy) = (2, 1)$ 时, 从在嵌套上的 x 方向上延伸的区域, 搜集 4×4 的 DC 值, 并且, 当 $(sx, sy) = (2, 2)$ 时, 从在嵌套上的 x 和 y 方向上延伸的区域, 集中 4×4 的 DC 值, 并存储到基底矢量 $\langle v_{nk} \rangle$ 的存储区域中。这是这样的处理: 利用原始图象具有的自相似性, 从嵌套上的 DC 图象以各种角度搜索与原始图象的像素块近似的图象。

图 5 和图 6 是实施例的适当正交化处理的流程图。在图 5 中, 在步骤 S31 中, 在误差寄存器 E_r 中设定较大的值 (例如, 100000), 并且, 把基底数寄存器 nk 初始化为「1」。在步骤 S32 中, 把嵌套图象的开始地址寄存器 x,y 同时初始化为「0」。在步骤 S33 中, 把嵌套图象的二次取样间隔寄存器 sx,sy 同时初始化为「1」。

在步骤 S34 中, 从嵌套图象取出 4×4 的 DC 块图象 $B_{y,x,sy,sx}$, 来制成基底矢量 $\langle v_{nk} \rangle$ {相当于后述的图 11 (C) 中的基底矢量 $\langle u_{nk} \rangle$ }, 并且, 在 $nk > 1$ 的情况下, 通过 GRAM SCHMIDT 正交化法, 来与其以前的基底矢量正交化。

所谓 GRAM SCHMIDT 正交化法是这样的方法: 由 n 维内积空间 V 的一个基底 $\{v_1, \dots, v_n\}$ 来构成 V 的正规正交基底 $\{v'_1, \dots, v'_n\}$, 以下参照图 11 (C) 来简要说明 gram schmidt 的正交化法。当为第一基底矢量 $\langle v_1 \rangle$ (其中, 相当于 $\langle u_1 \rangle$) 时, 第一正规化基底矢量 $\langle v'_1 \rangle$ 是单位矢量, 因此, 具有:

$$v'_1 = \frac{v_1}{\|v_1\|} = \alpha_{11} u_1$$

接着, 当从嵌套图象抽出第二基底矢量 $\langle u_2 \rangle$ 时, 假定将与上述第一正规化基底矢量 $\langle v'_1 \rangle$ 正交的第二基底矢量 $\langle u_2 \rangle$ 设置为:

$$v_2 = u_2 + k v_2$$

这样，从 $\langle v_2 \rangle \cdot \langle v'_1 \rangle = 0$ ，而得到这样的关系：

$$v_2 \cdot v'_1 = (u_2 + k v'_1) \cdot v'_1 = u_2 \cdot v'_1 + k (v'_1 \cdot v'_1) = u_2 \cdot v'_1 + k = 0$$

此时，标量系数 k 为：

$$k = - (u_2 \cdot v'_1)$$

而且，当把系数 k 代入上式时，第二正交基底矢量 $\langle v_2 \rangle$ 用下式表示：

$$v_2 = u_2 - (u_2 \cdot v'_1) v'_1$$

而且，第二正规化基底矢量 $\langle v'_2 \rangle$ 也是单位矢量，因此，具有：

$$v'_2 = \frac{v_2}{\|v_2\|} = \frac{u_2 - (u_2 \cdot v'_1) v'_1}{\|u_2 - (u_2 \cdot v'_1) v'_1\|} = \alpha_{21} u_1 + \alpha_{22} u_2$$

类似地，第 n 正规化基底矢量 $\langle v'_n \rangle$ 为：

$$v'_n = \frac{v_n}{\|v_n\|} = \frac{u_n - (u_n \cdot v'_1) v'_1 - \dots - (u_n \cdot v'_{n-1}) v'_{n-1}}{\|u_n - (u_n \cdot v'_1) v'_1 - \dots - (u_n \cdot v'_{n-1}) v'_{n-1}\|} = \sum_{i=1}^n \alpha_{ni} u_i$$

在步骤 S35 中，使用正规化基底矢量 $\langle v'_{nk} \rangle$ ，来求出与残差矢量 $\langle d \rangle$ 的距离成为最小的基底矢量的展开系数 α_{nk} 。

在图 12 (A) 的 (a) 中，表示使用正规化基底矢量 $\langle v'_{nk} \rangle$ 来近似残差矢量 $\langle d \rangle$ 的处理的图形。在图 12 (A) 的 (a) 中，残差矢量 $\langle d \rangle$ 与乘以了展开系数 α_{nk} 的基底矢量 $\alpha_{nk} \langle v'_{nk} \rangle$ 的差矢量 $\{ \langle d \rangle - \alpha_{nk} \langle v'_{nk} \rangle \}$ 的幅值为最小的，这是因为：基底矢量 $\alpha_{nk} \langle v'_{nk} \rangle$ 与差矢量 $\{ \langle d \rangle - \alpha_{nk} \langle v'_{nk} \rangle \}$ 正交时（内积=0），而求出符合下式的正规化基底矢量 $\langle v'_{nk} \rangle$ 的展开系数 α_{nk} 。

$$\begin{aligned} (d - \alpha_{nk} v'_{nk}) \cdot \alpha_{nk} v'_{nk} &= 0 \\ \alpha_{nk} d \cdot v'_{nk} - \alpha_{nk}^2 v'_{nk} \cdot v'_{nk} &= 0 \end{aligned}$$

$$\alpha_{nk} = \frac{d \cdot v'_{nk}}{\|v'_{nk}\|^2}$$

下面对图 12 (A) 的 (b) 进行说明。

5 返回图 5, 在步骤 S36 中, 求出用基底矢量 $\alpha_{nk} \langle v'_{nk} \rangle$ 近似残差矢量 $\langle d \rangle$ 时的误差矢量的幅值 ε_r 。在步骤 S37 中, 判别是否 $\varepsilon_r < E_r$ 。当 $\varepsilon_r < E_r$ 时, 在步骤 S38 中, 在用于存储与 ε_r 的最小值相关的各种信息的寄存器 E_r, X, Y, SX, SY 中分别保持此时的 $\varepsilon_r, x, y, sx, sy$ 。而且, 在寄存器 α 中存储此时的展开系数 α_{nk} , 在基底矢量的存储区域 $\langle v \rangle$ 中存储此时的基底矢量 $\langle v_{nk} \rangle$, 在正交化基底矢量的存储区域 $\langle v' \rangle$ 中存储此时的正交化基底矢量 $\langle v'_{nk} \rangle$ 。而且, 当不是 $\varepsilon_r < E_r$ 时, 跳过上述步骤 S38 的处理。

10

在步骤 S39 中, 使取样间隔 $sx+1$, 在步骤 S40 中, 判别是否 $sx > 2$ 。当不是 $sx > 2$ 时, 返回步骤 S34, 此时, 对以不同的取样间隔 sx 所抽出的基底矢量 $\langle v_{nk} \rangle$ 进行与上述相同的处理。以此类推, 也就是, 当在步骤 S40 的判别中是 $sx > 2$ 时, 在步骤 S41 中, 把 sx 初始化为「1」, 并且, 使取样间隔 $sy+1$, 在步骤 S42 中, 判别是否 $sy > 2$ 。当结果不是 $sy > 2$ 时, 返回步骤 S34, 此时, 对以不同的取样间隔 sy 所抽出的基底矢量 $\langle v_{nk} \rangle$ 进行与上述相同的处理。以此类推, 也就是, 当在步骤 S41 的判别中是 $sy > 2$ 时, 首先, 对于嵌套图象的开始位置 $(x,y) = (0, 0)$, 尝试具有不同的取样间隔 $(sx,sy) = (1, 1), (1, 2), (2, 1), (2, 2)$ 的共计 4 种的各基底矢量 $\langle v_{nk} \rangle$ 。处理进行到图 6 的步骤 S43。

15

20

25 在步骤 S43 中, 使嵌套上的开始位置 $x+1$, 在步骤 S44 中, 判别是否 $x > p$ (例如, $p=63$)。在不是 $x > p$ 的情况下, 返回步骤 S33 (②), 此时, 对在 x 方向上进行了 1DC 值移位的开始位置的各基底矢量 $\langle v_{nk} \rangle$ 进行与上述相同的处理。这样一来, 当在步骤 S44 的判别中是 $x > p$ 时, 在步骤 S45 中, 把开始位置 x 初始化为「0」, 并且,

30

在开始位置 $y+1$ 。在步骤 S46 中, 判别是否 $y>q$ (例如, $q=31$)。在不是 $y>q$ 的情况下, 返回步骤 S33 (②), 此时, 对在 y 方向上进行了 1DC 值移位的开始位置的各基底矢量 $\langle v_{nk} \rangle$ 进行与上述相同的处理。以此类推, 当在步骤 S46 的判别中是 $y>q$ 时, 对于嵌套图象的全部开始位置 $(x,y) \in [0, 63] \times [0, 31]$, 尝试全部取样间隔 $(sx,sy) \in (1, 1), (1, 2), (2, 1), (2, 2)$ 的各基底矢量 $\langle v_{nk} \rangle$ 。处理进到步骤 S47。

在步骤 S47 中, 从残差矢量 $\langle d \rangle$ 减去在此时刻造成最小的近似误差 ε_r 的正交基底矢量 $\alpha \langle v' \rangle$, 求出误差的幅值, 判别该幅值是否小于允许值 Z 。在不小于的情况下, 在步骤 S48 中, 用 $\{ \langle d \rangle - \alpha \langle v' \rangle \}$ 来置换残差矢量 $\langle d \rangle$, 进行更新。而且, 设在此时刻造成最小的近似误差 ε_r 的展开系数 α 、基底矢量 $\langle v \rangle$ 、正交基底矢量 $\langle v' \rangle$ 的内容为 α_{nk} 、 $\langle v_{nk} \rangle$ 、 $\langle v'_{nk} \rangle$ 退避・保持在存储区域中。因此, 当 $nk=1$ 时, 保存・保持 α_1 、 $\langle v_1 \rangle$ 、 $\langle v'_1 \rangle$ 。并且, 在 nk 上+1, 返回步骤 S32 (①)。

以此类推即, 当在步骤 S47 中的判别是近似残差小于允许值 Z 时, 进到步骤 S49, 把由各正交基底矢量 $\alpha_{nk} \langle v'_{nk} \rangle$ (其中, $nk=1, 2, \dots$) 的线性组合组成的矢量变换为由各基底矢量 $\beta_{nk} \langle \beta_{nk} \rangle$ {相当于图 11 (C) 的各基底矢量 $\langle u_{nk} \rangle$ } 的线性组合组成的矢量。以下把系数 β 称为基底矢量 $\langle v \rangle$ 的展开系数。

在此, 具体说明上述步骤 S49 的变换方法。当 $k=1 \sim nk$ 时, 当由基底矢量 $\langle v_k \rangle$ {其中, 相当于上述图 11 (C) 的基底矢量 $\langle u_{nk} \rangle$ } 组成的矩阵为 V , 由标量展开系数 β_k 组成的矩阵为 B , 由正规化正交基底矢量 $\langle v'_k \rangle$ 组成的矩阵为 V' , 由标量系数 α_k 组成的矩阵为 A 时, 把它们分别设置为:

$$V = [\nu_1, \nu_2, \dots, \nu_{nk}] \quad B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{nk} \end{bmatrix} \quad V' = [\nu'_1, \nu'_2, \dots, \nu'_{nk}] \quad A = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{nk} \end{bmatrix}$$

上述步骤 S49 的变换可以通过设置为: $VB=V' A$ 来得到。为了对矩阵 B 来求解其, 首先, 把矩阵 V 变换为正方矩阵, 把矩阵 V^T (V^T 是 V 的转置矩阵) 从左乘以矩阵 V , 而得到:

$$V^T V B = V^T V' A$$

该矩阵 ($V^T V$) 被展开为这样:

$$V^T V = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_{nk} \end{bmatrix} [\nu_1, \nu_2, \dots, \nu_{nk}] = \begin{bmatrix} \nu_1 \bullet \nu_1 & \nu_1 \bullet \nu_2 & \dots & \nu_1 \bullet \nu_{nk} \\ \nu_2 \bullet \nu_1 & \nu_2 \bullet \nu_2 & \dots & \nu_2 \bullet \nu_{nk} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{nk} \bullet \nu_1 & \nu_{nk} \bullet \nu_2 & \dots & \nu_{nk} \bullet \nu_{nk} \end{bmatrix}$$

其中, $\langle v_i \rangle \bullet \langle v_j \rangle$ 表示内积, 并且, 由于 $\langle v_i \rangle \bullet \langle v_j \rangle = \langle v_j \rangle \bullet \langle v_i \rangle$, 对于对角要素, 得到对称的正方矩阵, 并且, 由于 $\langle v_i \rangle$ 与 $\langle v_j \rangle$ 不相同, 而存在逆矩阵。因此, 进一步从左乘以矩阵 ($V^T V$) 的逆矩阵 ($V^T V$)⁻¹, 而得到:

$$(V^T V)^{-1} V^T V B = B = (V^T V)^{-1} V^T V' A$$

这样, 在本实施例中, 为使与残差矢量 $\langle d \rangle$ 的误差近似于小于允许值 Z , 在探索了所需要的最低限度个数 nk 的正交化基底矢量 $\langle v'_{nk} \rangle$ 之后, 用原来的基底矢量 $\langle v_{nk} \rangle$ ($= \langle u_{nk} \rangle$) 的线性结合来表示它们, 并且, 求出各基底矢量 $\langle v_{nk} \rangle$ 的展开系数 β_{nk} , 把它们进行编码, 通过这样的构成, 在解码时, 就不需要 GRAM SCHMIDT 的正交化计算。并且, 省略了把范数正规化为 1 的处理。

图 12 (A) 的 (b) 表示了基底数 $nk=3$ 时的残差矢量的近似图形。

最初，求出使与残差矢量 $\langle d \rangle$ 的误差 ε_r 成为最小的第一基底矢量 α_1 $\langle v'_1 \rangle$ 。接着，与该矢量 $\langle v'_1 \rangle$ 正交，并且求出使与所更新的其余的残差矢量 $\langle d' \rangle$ 的误差 ε_r 成为最小的第二正交基底矢量 α_2 $\langle v'_2 \rangle$ 。接着，与该矢量 $\langle v'_2 \rangle$ 正交，并且求出使与所更新的其余的残差矢量 $\langle d' \rangle$ 的误差 ε_r 成为最小的第三正交基底矢量 α_3 $\langle v'_3 \rangle$ 。

返回图 3，在步骤 S9 中，判别是否使用的基底数 $nk > 7$ 。当 $nk > 7$ (8 以上) 时，即使使用基底矢量也不会产生图象压缩的优点，因此，进到步骤 S19，将基底数「8」作为代码 F_1 ，将使用基底数「0」作为代码 F_2 ，以及将残差矢量 $\langle d \rangle$ 本身作为代码 F_3 ，分别输出。当 $nk \leq 7$ 时，进到步骤 S10，与上述基底矢量的使用一起来执行以下的标量量化处理。

在标量量化处理中，首先，通过下式来求出以所选择的 nk 个之内所选择的顺序采用各个 m 个 ($0 \leq m \leq nk$) 的基底时的误差矢量 $\langle d_m \rangle$ ：

$$d_m = d - \sum_{i=1}^m \beta_i v_i$$

并且， $m=0$ 时的 $\sum_{i=1}^m \beta_i v_i$ 代表零矢量。而且，用相当于没有被采用的其余基底数 $yk = nk - m$ 的代码量来对该误差矢量 $\langle d_m \rangle$ 进行标量量化。即，用与其余基底数 yk 相对应的量化系数 Q_{yk} 对误差矢量 $\langle d_m \rangle$ 进行标量量化，用 $[-2^{yk-1}, 2^{yk-1} - 1]$ 的值域来对该值限幅。

因此，这样标量量化·限幅的误差矢量 $\langle d'_m \rangle$ 在其解码后并不限于与原来的误差矢量 $\langle d_m \rangle$ 相同。因此，预先对各个 m ($m=1, 2, \dots, nk$)，用分别与各量化误差矢量 $\langle d'_m \rangle$ 相对应的量化系数 Q_{yk} 进行逆量化，来求出各逆量化误差矢量 $\langle d''_m \rangle$ ，寻找误差 $\| \langle d_m \rangle - \langle d''_m \rangle \parallel$ 成为最小的 m ，最终，生成采用该 m 个基底时的代码。

在图 8 (B) 中显示了标量量化表。

当 $nk=8$ 时，使用一个基底时 ($m=1$) 的其余基底数 $yk=7$ ，以此类推使用 7 个基底时 ($m=7$) 的其余基底数 $yk=1$ 。

5 而且，在后述中，在本实施例中，对于一个基底，在其代码中使用 2 字节，因此，当使用八个基底时，总的代码长度为 16 字节。但是，如果有 16 字节的容量，能够直接对 16 维的残差矢量 $\langle d \rangle$ 进行编码，因此，就没有进行适当正交化处理的质量。所以，在 0~7 个的范围内使用基底，据此，其余基底数 yk 为 7~1。

10 各量化系数 Q_{yk} 根据其余基底数 yk 而预先决定。优选地，对各量化系数 Q_{yk} 做这样的选择：把误差矢量 $\langle d_m \rangle$ 容纳在其余基底数 yk 的代码量内。

15 当其余基底数 $yk=1$ 时，可以在误差矢量 $\langle d_m \rangle$ 的标量量化中使用 16×1 比特。这是误差矢量 $\langle d_m \rangle$ 的每个要素中 1 比特的代码量。即，在此情况下，用 $Q_{yk}=8$ 来对误差矢量 $\langle d_m \rangle$ 进行标量量化，把其值限幅为 $[-2^{1-1}, 2^{1-1}-1] = [-1, 0]$ 。其中，例如，代码的比特「0」对应于幅值「-1」，比特「1」对应于幅值「0」。以此类推，当其余基底数 $yk=7$ 时，可以在误差矢量 $\langle d_m \rangle$ 的标量量化中使用 16×7 比特。这是误差矢量 $\langle d_m \rangle$ 的每个要素中 7 比特的代码量。在此情况下，用 $Q_{yk}=2$ 来对误差矢量 $\langle d_m \rangle$ 进行标量量化，把其值限幅为 $[-2^{7-1}, 2^{7-1}-1] = [-64, 63]$ 。这样，通过同时使用 m 个基底和标量量化的误差，能够不改变代码量而提高图象质量。由于不需要其余基底数 yk 部分的展开系数 β ，则减少了若干代码量。

30 图 7 是标量量化处理的流程图。在步骤 S61 中，把使用基底数计数用的寄存器 m 初始化为「0」。在步骤 S62 中，求出采用 m 个基底后的误差矢量 $\langle d_m \rangle$ 。而且， $m=0$ 时的 Σ 项表示零矢量。即，误差矢量 $\langle d_0 \rangle = \langle d \rangle$ (残差矢量)。在步骤 S63 中，通过 nk ($1 \leq nk \leq 7$)

个之中的这次的尝试, 用 $y_k = nk - m$ 来求出未被使用的其余基底数 y_k 。在步骤 S64 中, 用 Q_{y_k} 对误差矢量 $\langle d_m \rangle$ 进行标量量化, 将所得结果的误差矢量 $\langle d'_m \rangle$ 的各元素限幅在 $[-2^{y_k-1}, 2^{y_k-1}-1]$ 的范围内。

5 在步骤 S65 中, 用相同的 Q_{y_k} 对上述量化 (并且限幅) 后的误差矢量 $\langle d'_m \rangle$ 进行标量逆量化。在步骤 S66 中, 求出把原来的误差矢量 $\langle d_m \rangle$ 与上述逆量化的误差矢量 $\langle d'_m \rangle$ 进行比较时的误差 ε_{r_m} , 将其保存在预定区域内。在步骤 S67 中, 在寄存器 m 上 +1。在步骤 S68 中, 判别是否 $m > nk$ ($1 \leq nk \leq 7$)。当不是 $m > nk$ 时, 返回步骤

10 S62, 进行上述同样的处理。

 在上述步骤 S68 的判别中, 当 $m > nk$ 时, 求出采用各 m ($m=0, 1, 2, 3, \dots, nk$) 个基底时的误差 $\varepsilon_{r_0} \sim \varepsilon_{r_{nk}}$ 。在步骤 S69 中, 从误差 $\varepsilon_{r_0} \sim \varepsilon_{r_{nk}}$ 中抽出最小的误差 $\varepsilon_{r_{mk}}$ 。在步骤 S70 中, 将基底数 nk 作为代码 F1 输出。在步骤 S71 中, 将使用基底数 (有效基底数) mk 作为代码 F2 输出。在步骤 S72 中, 把用 Q (例如 8) 对各使用基底的嵌套上的开始位置 (x, y) 、二次取样间隔 (sx, sy) 以及各使用基底的展开系数 β 进行量化的剩余在每个基底中压缩为 16 比特, 而作为代码 F_4 输出。

15 输出。

20 代码 F_4 的细目为: 在嵌套上的开始位置 (x, y) 中的横 6 比特和纵 5 比特的共计 11 比特、4 种之中的任一个二次取样间隔中的 2 比特、在展开系数 β 的剩余中的 3 比特, 总共 16 比特。这样, 代码 F_4 在每个使用基底中占用 2 字节。

25 在步骤 S73 中, 将用 Q (例如 8) 对上述基底的展开系数 β 进行量化的商作为代码 F_5 而输出。在步骤 S74 中, 把成为上述最小的逆量化误差 $\varepsilon_{r_{mk}}$ 时的误差矢量 $\langle d'_{mk} \rangle$ 分配到相当于其余基底数 y_k 的 $16 \times (nk - mk)$ 比特的区域中, 而作为代码 F_3 输出。接着, 退出该处理。

30 理。

返回图 3，在此时刻，原始图象的残差矢量 $\langle d \rangle$ 的编码结束。
 而且，在步骤 S11 中，使地址寄存器 $i+4$ ，在步骤 S12 中，判别是否
 $i > n$ 。当不是 $i > n$ 时，返回步骤 S5，对于下一个离开在 i 轴方向上偏
 移了 4 比特的位置的 1 残差矢量 $\langle d \rangle$ ，进行上述相同的处理，进行
 编码。以此类推，在上述步骤 S12 的判别中，当 $i > n$ 时，在步骤 S13
 中，把 i 初始化为「0」，并且，使 $j+4$ 。在步骤 S14 中，判别是否 j
 $> m$ ，当不是 $j > m$ 时，返回步骤 S5，对于下一个在 j 轴方向上偏移了
 4 比特的残差矢量 $\langle d \rangle$ ，进行上述相同的处理，进行编码。以此类推
 即，在上述步骤 S14 的判别中，当 $j > m$ 时，对全部图象的残差矢量
 $\langle d \rangle$ 的上述编码处理结束。

在图 12 (B) 中表示图象压缩编码的表。在以下的说明中，参照
 该表。

返回图 3，在步骤 S15 中，对可变长度代码 F_1 ， F_2 ， F_5 进行霍夫
 曼编码，并输出。但是，对于代码 F_1 的基底数「 nk 」，仅在 $nk=0$ 时，
 考虑扫描宽度，来进行霍夫曼编码。对于代码 F_2 的使用基底数「 mk 」，
 同样进行考虑。对于代码 F_5 ，对用常数 Q （例如 8）对展开系数 β 进
 行量化的商进行霍夫曼编码。而且，在其余基底数 $yk \neq 0$ 的情况下，
 写入用于表示象素块的切换的代码 EOB。

在步骤 S16 中，对可变长度代码 F_6 进行霍夫曼编码，并输出。
 代码 F_6 是用量化系数 Q_s 对由 DC 值的 2 维 DPCM 所产生的预测残差
 $\Delta DC_{j,l}$ 进行量化的。但是，仅在预测残差 $\Delta DC_{j,l}=0$ 时，考虑扫描宽
 度，而分别独立地对预测残差 $\Delta DC_{j,l}$ 和扫描宽度进行霍夫曼编码。

在步骤 S17 中，对可变长度代码 F_3 ，固定长度代码 F_4 进行霍夫
 曼编码，并输出。代码 F_3 是用 Q_{yk} 对使用基底数 $mk > 0$ 时的最终误差
 矢量 $\langle d'_{mk} \rangle$ 进行标量量化 $[\langle d'_{mk} \rangle / Q_{yk}]$ 的代码，当使用基底数 $mk=0$

时，为原始的残差矢量 $\langle d \rangle$ 的代码。代码 F_4 在每个基底中由嵌套的开始位置 (x,y) 的 11 比特、二次取样系数 (sx, sy) 的 2 比特、在展开系数 β 的剩余 (β/Q) 的 3 比特，总共 16 比特固定组成，它们按使用基底的出现顺序一个接一个地构成。

5

而且，作为上述代码的全体，以像素块为单位，按出现顺序一个接一个地构成代码串。实际上，在大半的块中，基底数 nk 变成为 2 个左右，并且所得到的少量可变长度代码仅是作为用 Q 对基底的展开系数 β 进行量化的商和 DC 图象的 DPCM 代码以及基底数 $\lceil nk \rceil$ ，以及所使用的基底数 $\lceil mk \rceil$ 本身。

10

图 13 是实施例的图象解码处理的流程图。在步骤 S81 中，读入图象编码数据。在步骤 S82 中，对 Y, U, V 的各 DC 值进行解冻（解码）。因此，通过 $DC'_{J,I} = DC'_{J,I} + [\Delta DC_{J,I}/Q_s]Q_s$ ，来得到解码结果的 DC 值 $DC'_{J,I}$ 。在此， $DC'_{J,I}$ 是解码侧的 DC 的预测值，由 $DC'_{J,I} = (DC'_{J,I-1} + DC'_{J-1,I})/2$ 来提供。在步骤 S83 中，从 Y 成分的 DC 值来生成嵌套。嵌套可以通过另外收取 DC 图象上的开始位置和尺寸的信息来生成。在步骤 S84 中，把解码（再现）图象的地址寄存器 i, j 共同初始化为 $\lceil 0 \rceil$ 。

15

20

在步骤 S85 中，输入与块图象（即残差矢量）相关的编码数据。在步骤 S86 中，判别是否基底数 $nk > 0$ 。当 $nk = 0$ 时，在步骤 S97 中，通过后述的不分段的交流成分预测法来求出 16 像素的亮度数据。当 $nk > 0$ 时，进一步在步骤 S87 中判别是否使用基底数 $mk > 0$ 。

25

当 $mk = 0$ 时，在步骤 S96 中，对残差矢量 $\langle d \rangle$ 进行逆量化。在步骤 S90 中，把解码 DC 与求得的残差矢量 $\langle d \rangle$ 相加。当 $mk > 0$ 时，在步骤 S88 中，求出其余基底数 $yk (= nk - mk)$ ，用 Q_{yk} 对误差矢量 $\langle d'_{mk} \rangle$ 进行逆量化。在步骤 S89 中，从符合各基底的开始位置 (x,y) 、二次取样间隔 (sx, sy) 的嵌套来制作 mk 个基底矢量，取与展开系数

30

B 的积，来形成由它们的线性组合构成的近似矢量（正交基底系统），误差矢量 $\langle d'_{mk} \rangle$ 与其进行合成，而再现原来的残差矢量 $\langle d \rangle$ 。在步骤 S90 中，把解码 DC 值与求出的残差矢量 $\langle d \rangle$ 相加。这样，通过上述任一种方法来再现 4×4 的块图象 $T_{j,i}$ 。在步骤 S91 中，把再现图象 $T_{j,i}$ 存储在图象存储器中。

在步骤 S92 中，使地址寄存器 $i+4$ ，在步骤 S93 中，判别是否 $i > n$ 。当不是 $i > n$ 时，返回步骤 S85，此时，对下一列的块图象数据 $T_{j,i}$ 进行解码，并存储在图象存储器中。以此类推，在步骤 S93 的判别中，当 $i > n$ 时，在步骤 S94 中，把 i 初始化为「0」，并且使 $j+4$ 。在步骤 S95 中，判别是否 $j > m$ ，当不是 $j > m$ 时，则返回步骤 S85，此时，对下一行的各块图象数据 $T_{j,i}$ 进行解码，并将其存储在图象存储器中。以此类推即，在步骤 S93 的判别中，当 $i > n$ 时，就结束图象解码处理。

而且，虽未图示，将从 YUV 系统变换为 RGB 系统。此时，在 U,V 的上取样中，不利用插补滤波器，把相同的值代入横向 2 像素。

图 14 是实施例的不分段的交流成分预测处理的图形。在本实施例中，通过不分段的交流成分预测法来从附近的 DC 图象一举生成原始像素块的近似图象（AC 图象）。

因此，根据上述现有的分段的交流成分预测法，通过下式从包含注目块的周围 4 块的各 DC 值（S,U,R,B,L）推定有关的块 S 上的小块 $S_1 \sim S_4$ 的各 DC 值：

$$S_1 = S + (U + L - B - R) / 8$$

$$S_2 = S + (U + R - B - L) / 8$$

$$S_3 = S + (B + L - U - R) / 8$$

$$S_4 = S + (B + R - U - L) / 8$$

在图 14 (A) 中再次表示了与上述图 17 (B) 相同的图。同样，在该第一段中，推定 $U_1 \sim U_4$, $L_1 \sim L_4$, $R_1 \sim R_4$, $B_1 \sim B_4$ 等。而且，通过循环使用上述方法，通过下式推定出 S_1 上的 4 象素 $P_1 \sim P_4$ 的各象素值：

$$P_1 = S_1 + (U_3 + L_2 - S_3 - S_2) / 8$$

$$P_2 = S_1 + (U_3 + S_2 - S_3 - L_2) / 8$$

$$P_3 = S_1 + (S_3 + L_2 - U_3 - S_2) / 8$$

$$P_4 = S_1 + (S_3 + S_2 - U_3 - L_2) / 8$$

通过下式推定出小块 S_2 上的 4 象素 $P_1 \sim P_4$ 的各象素值：

$$P_1 = S_2 + (U_4 + S_1 - S_4 - R_1) / 8$$

$$P_2 = S_2 + (U_4 + R_1 - S_4 - S_1) / 8$$

$$P_3 = S_2 + (S_4 + S_1 - U_4 - R_1) / 8$$

$$P_4 = S_2 + (S_4 + R_1 - U_4 - S_1) / 8$$

通过下式推定出小块 S_3 上的 4 象素 $P_1 \sim P_4$ 的各象素值：

$$P_1 = S_3 + (S_1 + L_4 - B_1 - S_4) / 8$$

$$P_2 = S_3 + (S_1 + S_4 - B_1 - L_4) / 8$$

$$P_3 = S_3 + (B_1 + L_4 - S_1 - S_4) / 8$$

$$P_4 = S_3 + (B_1 + S_4 - S_1 - L_4) / 8$$

通过下式推定出小块 S_4 上的 4 象素 $P_1 \sim P_4$ 的各象素值：

$$P_1 = S_4 + (S_2 + S_3 - B_2 - R_3) / 8$$

$$P_2 = S_4 + (S_2 + R_3 - B_2 - S_3) / 8$$

$$P_4 = S_4 + (B_2 + S_3 - S_2 - R_3) / 8$$

$$P_3 = S_4 + (B_2 + R_3 - S_2 - S_3) / 8$$

图 14 (B) 表示本实施例的不分段的交流成分预测法。在以下的说明中，参照图 14 (A)。首先，当求出 S_1 上的 4 象素 $P_1 \sim P_4$ 时，进

行 $S_2 \approx S_3 \approx S$, $U_3 \approx U$, $L_2 \approx L$ 的各个近似。当把该近似用于上述 S_1 上的 P_1 的式中时, 得到:

$$\begin{aligned} P_1 &= S_1 + (U_3 + L_2 - S_3 - S_2) / 8 \\ &= S_1 + (U + L - S - S) / 8 \end{aligned}$$

5

而且, 当把上述 S_1 的式 $S_1 = S + (U + L - B - R) / 8$ 代入该式时, S_1 上的 P_1 最终用: $P_1 = S + (2U + 2L - 2S - B - R) / 8$ 来表示。

并且, 对于上述 S_1 上的 P_2 , 得到:

10

$$\begin{aligned} P_2 &= S_1 + (U_3 + S_2 - S_3 - L_2) / 8 \\ &= S_1 + (U + S - S - L) / 8 \end{aligned}$$

进而, 当把上述 S_1 的式 $S_1 = S + (U + L - B - R) / 8$ 代入该式时, S_1 上的 P_2 最终用: $P_2 = S + (2U - B - R) / 8$ 来表示。

15

并且, 对于上述 S_1 上的 P_3 , 得到:

$$\begin{aligned} P_3 &= S_1 + (S_3 + L_2 - U_3 - S_2) / 8 \\ &= S_1 + (S + L - U - S) / 8 \end{aligned}$$

20

进而, 当把上述 S_1 的式 $S_1 = S + (U + L - B - R) / 8$ 代入该式时, S_1 上的 P_3 最终用: $P_3 = S + (2L - B - R) / 8$ 来表示。

并且, 对于上述 S_1 上的 P_4 , 得到:

$$\begin{aligned} P_4 &= S_1 + (S_3 + S_2 - U_3 - L_2) / 8 \\ &= S_1 + (S + S - U - L) / 8 \end{aligned}$$

25

进而, 当把上述 S_1 的式 $S_1 = S + (U + L - B - R) / 8$ 代入该式时, S_1 上的 P_4 最终用: $P_4 = S + (2S - B - R) / 8$ 来表示。

30

这样, S_1 上的 $P_1 \sim P_4$ 最终用下式表示:

$$P_1 = S + (2U + 2L - 2S - B - R) / 8$$

$$P_2 = S + (2U - B - R) / 8$$

$$P_3 = S + (2L - B - R) / 8$$

$$P_4 = S + (2S - B - R) / 8$$

5

接着，当求出 S_2 上的 4 象素 $P_1 \sim P_4$ 时，进行 $S_1 \approx S_4 \approx S$ ， $R_1 \approx R$ ， $U_4 \approx U$ 的各个近似。当把该近似用于上述 S_2 上的 P_1 的式中时，得到：

$$\begin{aligned} P_1 &= S_2 + (U_4 + S_1 - S_4 - R_1) / 8 \\ &= S_2 + (U + S - S - R) / 8 \end{aligned}$$

10

而且，当把上述 S_2 的式 $S_2 = S + (U + R - B - L) / 8$ 代入该式时， S_2 上的 P_1 最终用： $P_1 = S + (2U - B - L) / 8$ 来表示。

并且，对于上述 S_2 上的 P_2 ，得到：

15

$$\begin{aligned} P_2 &= S_2 + (U_4 + R_1 - S_4 - S_1) / 8 \\ &= S_2 + (U + R - S - S) / 8 \end{aligned}$$

进而，当把上述 S_2 的式 $S_2 = S + (U + R - B - L) / 8$ 代入该式时， S_2 上的 P_2 最终用： $P_2 = S + (2U + 2R - 2S - B - L) / 8$ 来表示。

20

并且，对于上述 S_2 上的 P_3 ，得到：

$$\begin{aligned} P_3 &= S_2 + (S_4 + S_1 - U_4 - R_1) / 8 \\ &= S_2 + (S + S - U - R) / 8 \end{aligned}$$

25

进而，当把上述 S_2 的式 $S_2 = S + (U + R - B - L) / 8$ 代入该式时， S_2 上的 P_3 最终用： $P_3 = S + (2S - B - L) / 8$ 来表示。

并且，对于上述 S_2 上的 P_4 ，得到：

$$\begin{aligned} P_4 &= S_2 + (S_4 + R_1 - U_4 - S_1) / 8 \\ &= S_2 + (S + R - U - S) / 8 \end{aligned}$$

30

进而，当把上述 S_2 的式 $S_2 = S + (U + R - B - L) / 8$ 代入该式时， S_2 上的 P_4 最终用： $P_4 = S + (2R - B - L) / 8$ 来表示。

5 这样， S_2 上的 $P_1 \sim P_4$ 最终用下式表示：

$$P_1 = S + (2U - B - L) / 8$$

$$P_2 = S + (2U + 2R - 2S - B - L) / 8$$

$$P_3 = S + (2S - B - L) / 8$$

$$P_4 = S + (2R - B - L) / 8$$

10

接着，当求出 S_3 上的 4 像素 $P_1 \sim P_4$ 时，进行 $S_1 \approx S_4 \approx S$ ， $L_4 \approx L$ ， $B_1 \approx B$ 的各个近似。当把该近似用于上述 S_3 上的 P_1 的式中时，得到：

$$\begin{aligned} P_1 &= S_3 + (S_1 + L_4 - B_1 - S_4) / 8 \\ &= S_3 + (S + L - B - S) / 8 \end{aligned}$$

15

而且，当把上述 S_3 的式 $S_3 = S + (B + L - U - R) / 8$ 代入该式时， S_3 上的 P_1 最终用： $P_1 = S + (2L - U - R) / 8$ 来表示。

并且，对于上述 S_3 上的 P_2 ，得到：

$$\begin{aligned} P_2 &= S_3 + (S_1 + S_4 - B_1 - L_4) / 8 \\ &= S_3 + (S + S - B - L) / 8 \end{aligned}$$

20

进而，当把上述 S_3 的式 $S_3 = S + (B + L - U - R) / 8$ 代入该式时， S_3 上的 P_2 最终用： $P_2 = S + (2S - U - R) / 8$ 来表示。

25

并且，对于上述 S_3 上的 P_3 ，得到：

$$\begin{aligned} P_3 &= S_3 + (B_1 + L_4 - S_1 - S_4) / 8 \\ &= S_3 + (B + L - S - S) / 8 \end{aligned}$$

30 进而，当把上述 S_3 的式 $S_3 = S + (B + L - U - R) / 8$ 代入该式时， S_3

上的 P_3 最终用: $P_3 = S + (2B + 2L - 2S - U - R) / 8$ 来表示。

并且, 对于上述 S_3 上的 P_4 , 得到:

$$\begin{aligned} P_4 &= S_3 + (B_1 + S_4 - S_1 - L_4) / 8 \\ &= S_3 + (B + S - S - L) / 8 \end{aligned}$$

进而, 当把上述 S_3 的式 $S_3 = S + (B + L - U - R) / 8$ 代入该式时, S_3 上的 P_4 最终用: $P_4 = S + (2B - U - R) / 8$ 来表示。

这样, S_3 上的 $P_1 \sim P_4$ 最终用下式表示:

$$\begin{aligned} P_1 &= S + (2L - U - R) / 8 \\ P_2 &= S + (2S - U - R) / 8 \\ P_3 &= S + (2B + 2L - 2S - U - R) / 8 \\ P_4 &= S + (2B - U - R) / 8 \end{aligned}$$

接着, 当求出 S_4 上的 4 象素 $P_1 \sim P_4$ 时, 进行 $S_2 \approx S_3 \approx S$, $R_3 \approx R$, $B_2 \approx B$ 的各个近似。当把该近似用于上述 S_4 上的 P_1 的式中时, 得到:

$$\begin{aligned} P_1 &= S_4 + (S_2 + S_3 - B_2 - R_3) / 8 \\ &= S_4 + (S + S - B - R) / 8 \end{aligned}$$

而且, 当把上述 S_4 的式 $S_4 = S + (B + R - U - L) / 8$ 代入该式时, S_4 上的 P_1 最终用: $P_1 = S + (2S - U - L) / 8$ 来表示。

并且, 对于上述 S_4 上的 P_2 , 得到:

$$\begin{aligned} P_2 &= S_4 + (S_2 + R_3 - B_2 - S_3) / 8 \\ &= S_4 + (S + R - B - S) / 8 \end{aligned}$$

进而, 当把上述 S_4 的式 $S_4 = S + (B + R - U - L) / 8$ 代入该式时, S_4 上的 P_2 最终用: $P_2 = S + (2R - U - L) / 8$ 来表示。

并且，对于上述 S_4 上的 P_3 ，得到：

$$\begin{aligned} P_3 &= S_4 + (B_2 + S_3 - S_2 - R_3) / 8 \\ &= S_4 + (B + S - S - R) / 8 \end{aligned}$$

5 进而，当把上述 S_4 的式 $S_4 = S + (B + R - U - L) / 8$ 代入该式时， S_4 上的 P_3 最终用： $P_3 = S + (2B - U - L) / 8$ 来表示。

并且，对于上述 S_4 上的 P_4 ，得到：

$$\begin{aligned} P_4 &= S_4 + (B_2 + R_3 - S_2 - S_3) / 8 \\ 10 \quad &= S_4 + (B + R - S - S) / 8 \end{aligned}$$

进而，当把上述 S_4 的式 $S_4 = S + (B + R - U - L) / 8$ 代入该式时， S_4 上的 P_4 最终用： $P_4 = S + (2B + 2R - 2S - U - L) / 8$ 来表示。

15 这样， S_4 上的 $P_1 \sim P_4$ 最终用下式表示：

$$\begin{aligned} P_1 &= S + (2S - U - L) / 8 \\ P_2 &= S + (2R - U - L) / 8 \\ P_3 &= S + (2B - U - L) / 8 \\ P_4 &= S + (2B + 2R - 2S - U - L) / 8 \end{aligned}$$

20 这样一来，根据本实施例，从包含自己的附近的各 DC 值 (S, U, R, B, L) 直接（不分段）地求出 4×4 象素的近似象素。

25 图 15 是实施例的编码效率的曲线图。使用取样图象（ 320×240 象素，1 象素的 RGB 各成分为 8 比特精度），用 BPP（Bit Per Pixel 每象素的比特）、与 $BPR = [\text{压缩后的总数据量（比特）}] / [\text{原图象的象素数}]$ 相对应的 PSNR（峰-峰信噪比）、 $PSNR[dB] = 20 \log_{10} (255 / \sqrt{\epsilon^2})$ ，来评价编码性能。其中， ϵ^2 表示每个象素的均方误差。

30 图 15 (A) 表示使用动画图象的情况，实线表示本实施例的情况，

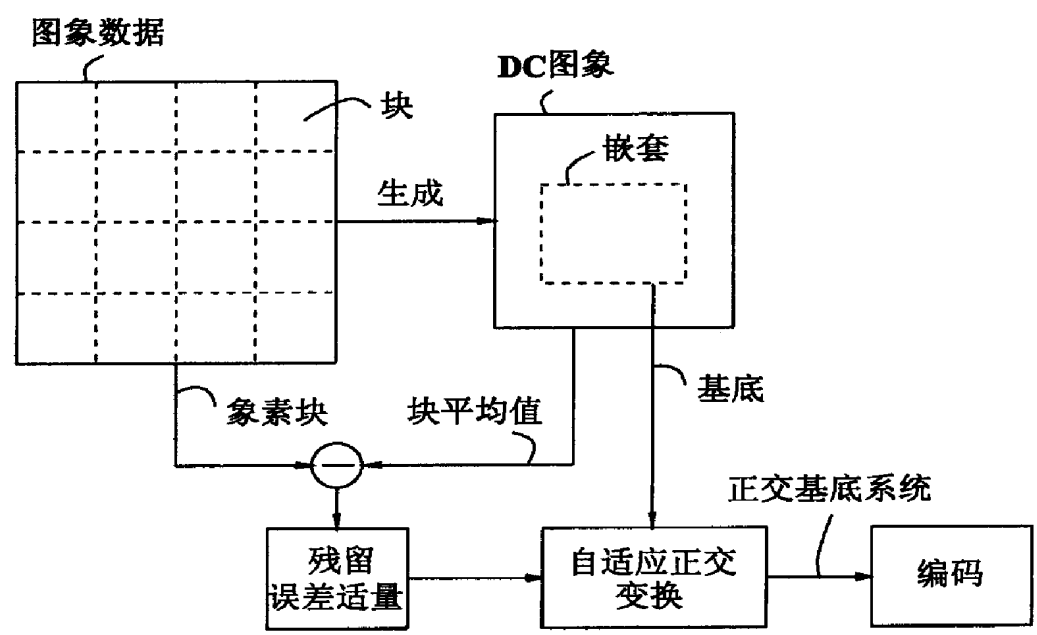
虚线表示 JPEG 的情况，如图所示的那样，得到大大高于 JPEG 的效率的编码性能。图 15 (B) 是使用 CG 图象的情况，得到与上述相同的倾向。但是，在本方式中，没有见到在正交变换编码法中所特有的蚊噪声和块噪声，则认为本方式对人工图象是有效的。虽未图示，对于自然图象也能得到与 JPEG 大致相同的性能。

虽然在上述实施例中描述了适用对游戏系统的例子，但是，本发明也可以用于照片图象、动画图象等通常的帧图象的编码/解码。

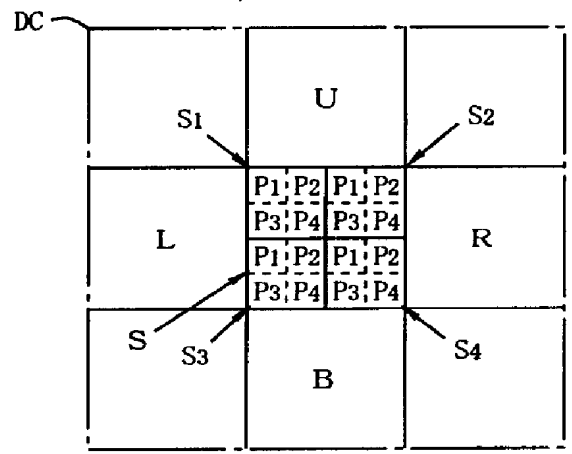
再有，虽然描述了适用于上述本发明的实施例，但是，在不脱离本发明思想的范围内，可以进行各个部分的构成、处理以及它们的组合的各种变更。

如上述那样，根据本发明，使嵌套轻量化，并且，能够削减可变长度代码量，因此，能够大幅度减轻解码时的计算负荷，即使在性能比较低的系统中，也能在不降低图象质量的情况下利用活动图象和静止图象的图象数据。而且，即使在轮廓部分也不会发生蚊噪声，而能够提供文字和动画图象等的再现性良好的高压缩率的图象数据。

说明书附图

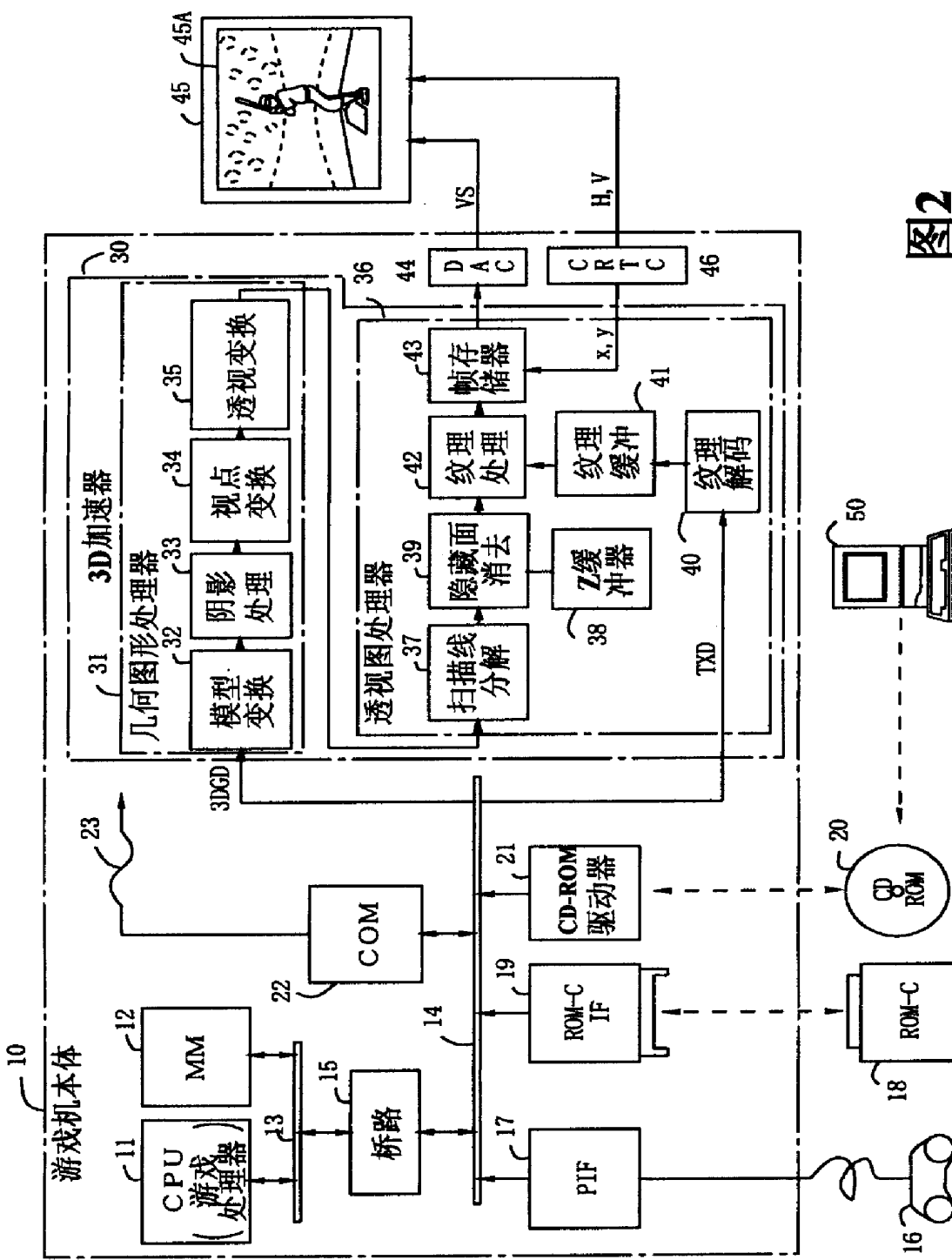


(A)



(B)

图1



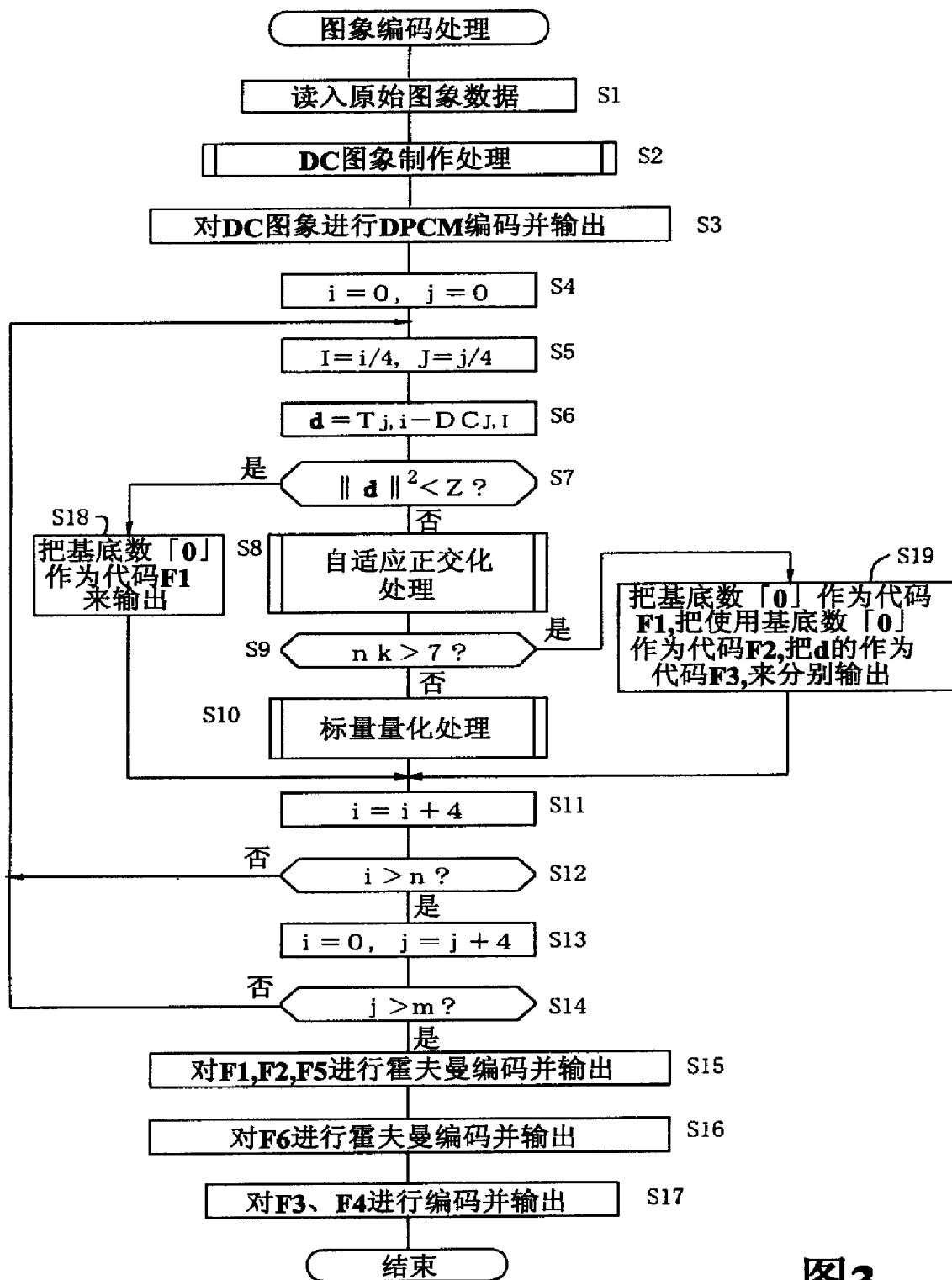


图3

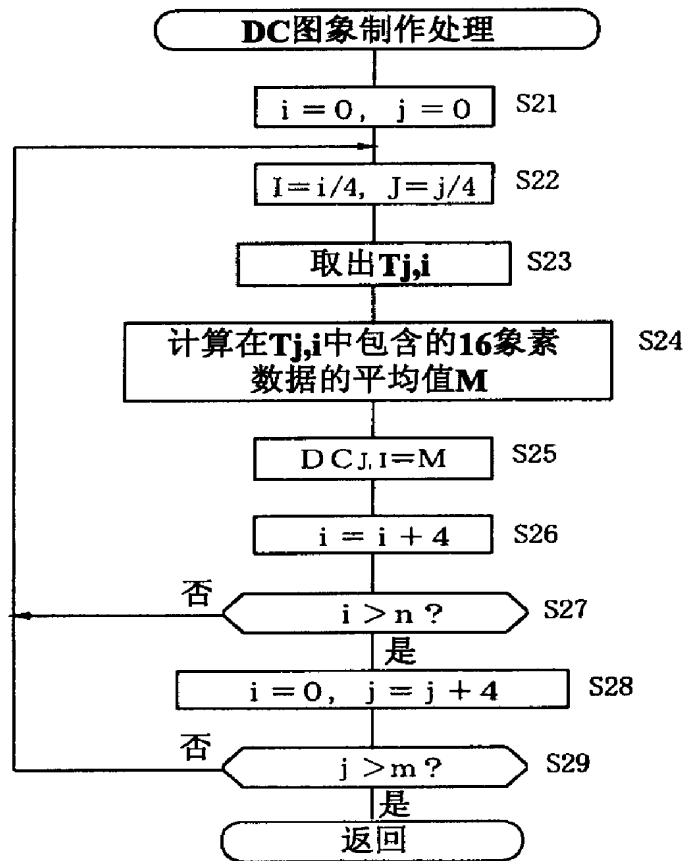


图4

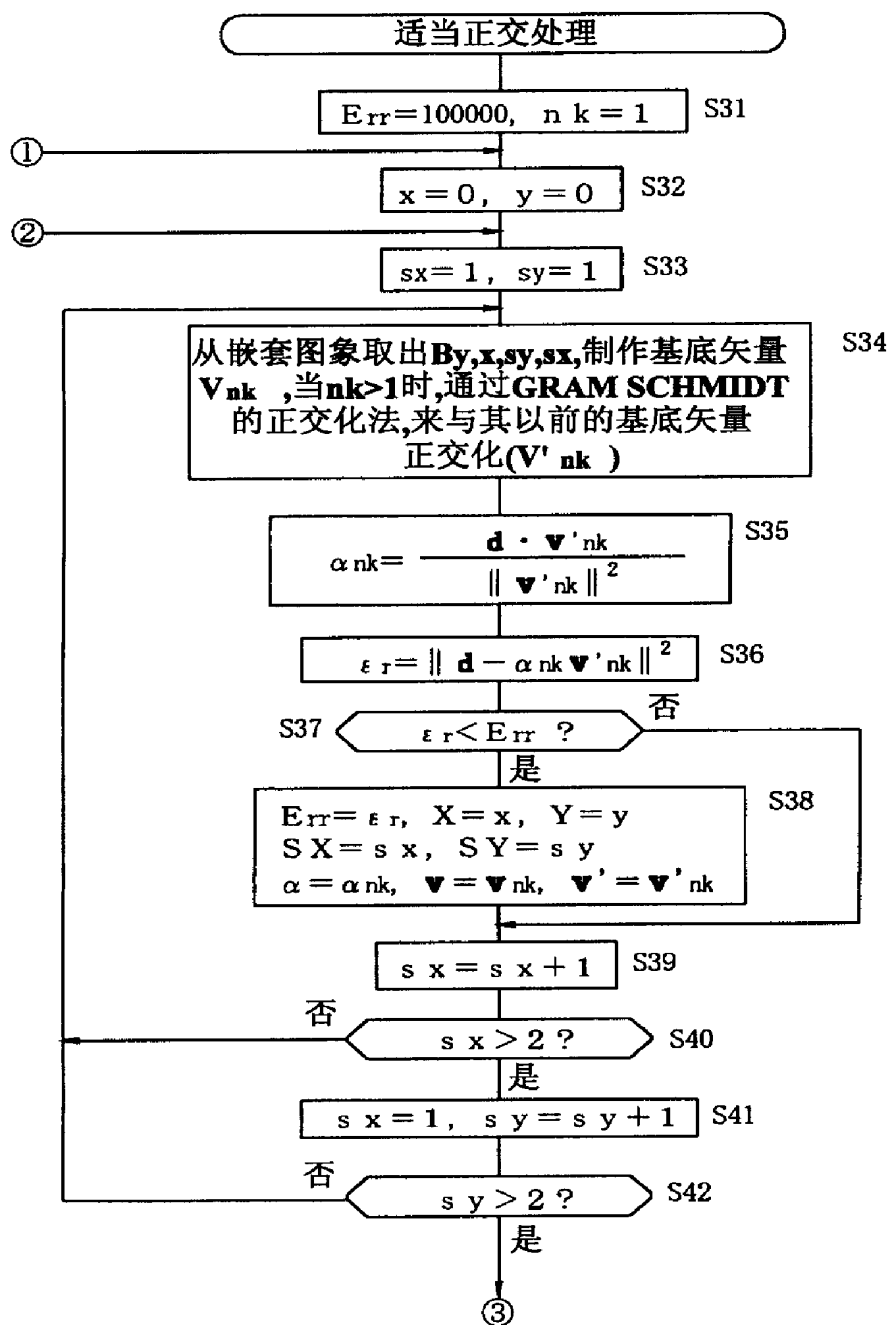


图5

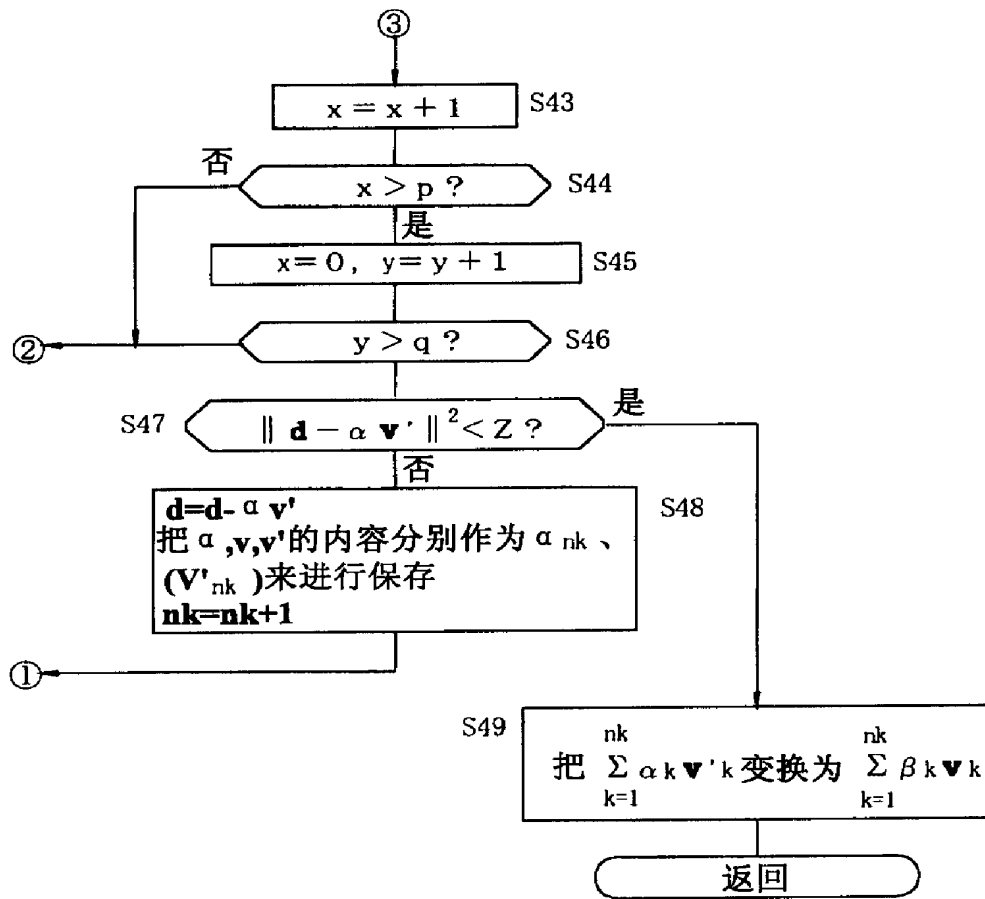


图6

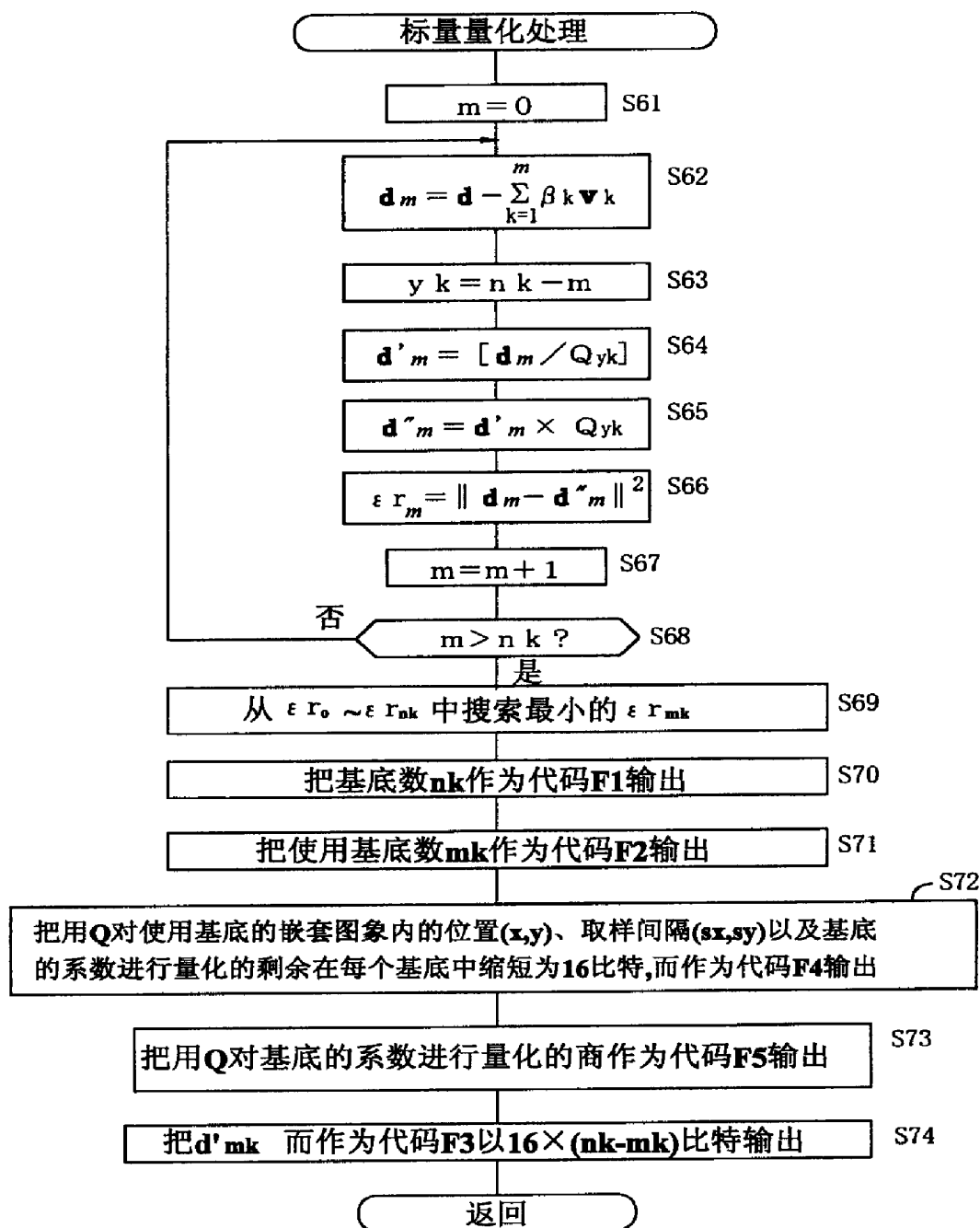


图7

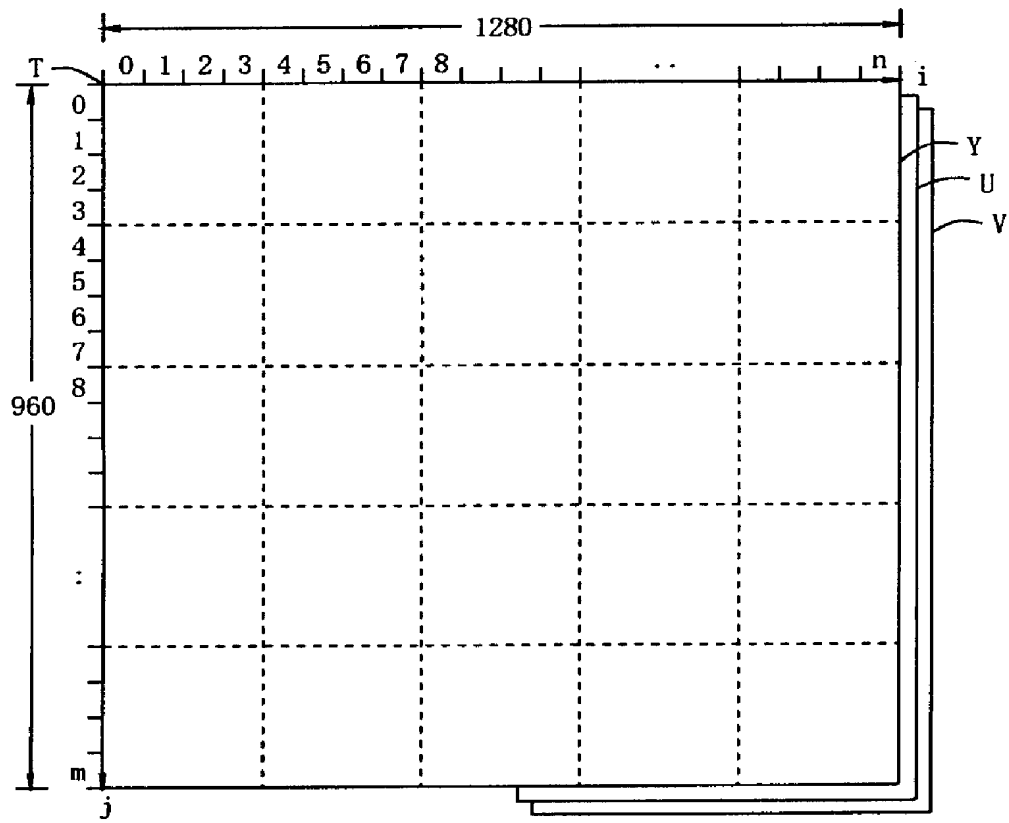
Z	Q_s
$Z \leq 100$	1
$100 < Z \leq 400$	2
$400 < Z \leq 1600$	4
$1600 < Z$	8

(A) 量化系数表

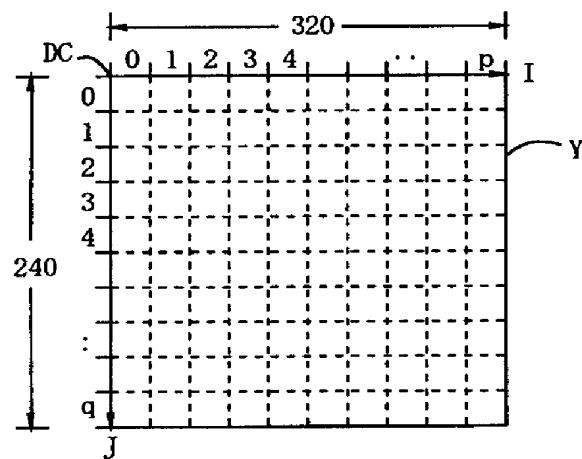
其余基底数 (y_k)	量化系数 (Q_{yk})	固定长度 代码(位)	代码的值域
7	2	16×7	$[-64, 63]$
6	3	16×6	$[-32, 31]$
5	4	16×5	$[-16, 15]$
4	5	16×4	$[-8, 7]$
3	6	16×3	$[-4, 3]$
2	7	16×2	$[-2, 1]$
1	8	16×1	$[-1, 0]$

(B) 标量量化表

图8

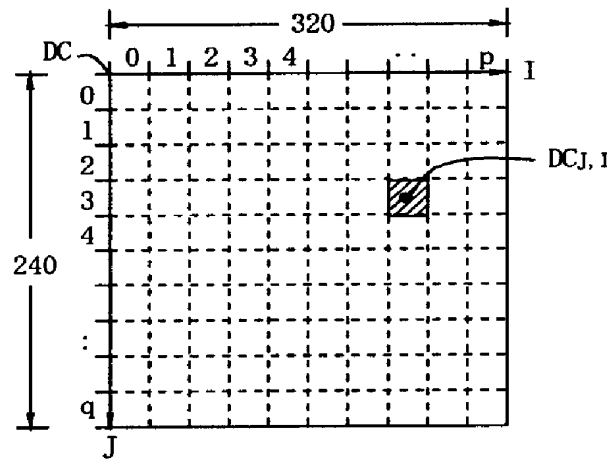


(A) 原图象

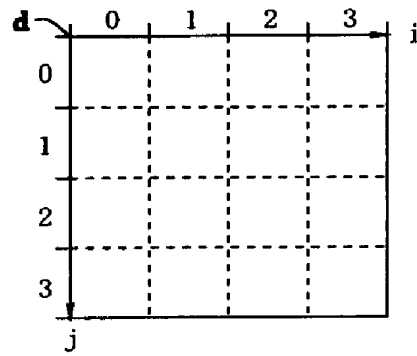


(B) DC图象

图9

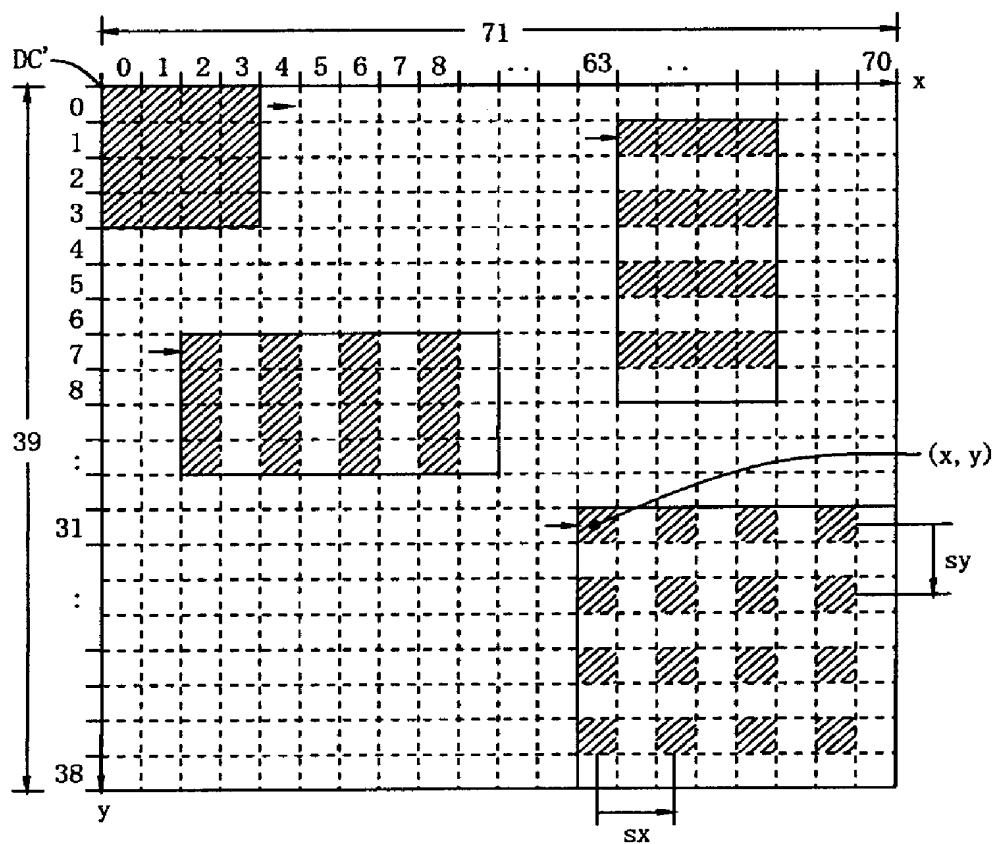


(A) DPCM编码

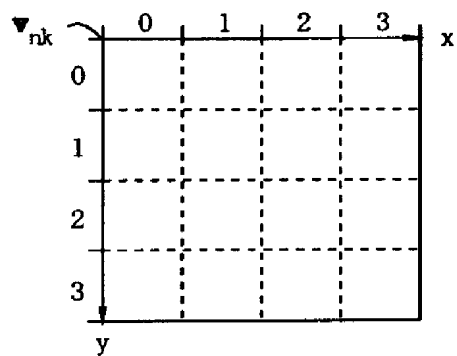


(B) 残留误差矢量

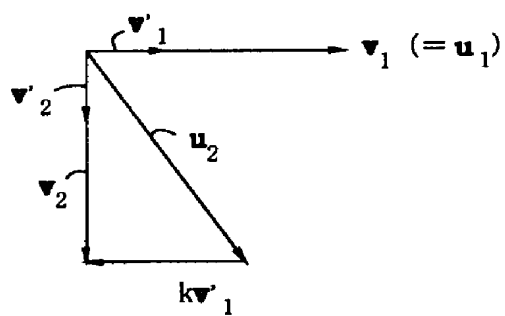
图10



(A) 嵌套数据

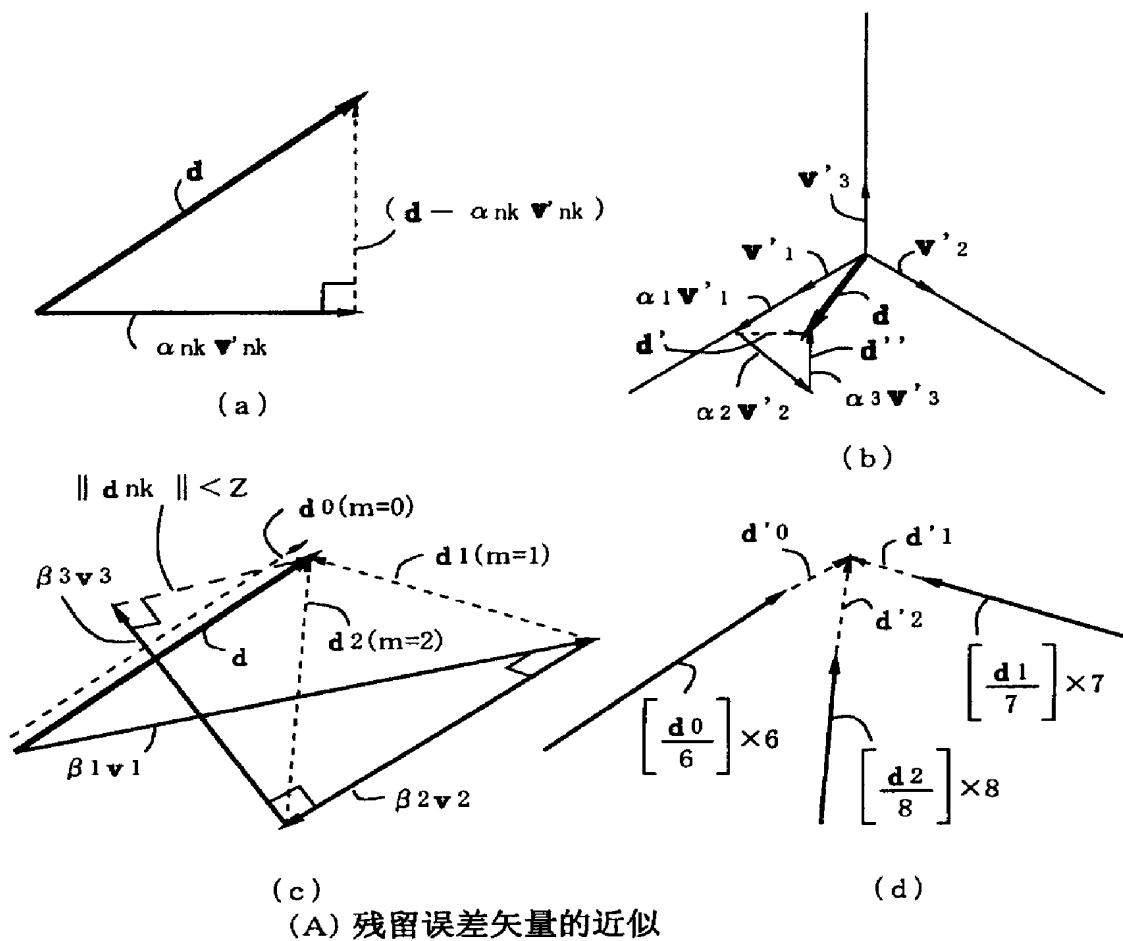


(B) 基底数据



(C) GRAM SCHMIDT 正交化

图11



(A) 残留误差矢量的近似

	$\ \mathbf{d} \ ^2 < Z$	$n_k \leq 7$	$n_k > 7$
F1	基底数「0」	基底数「 n_k 」	基底数「8」
F2		使用基底数「 n_k 」	使用基底数「0」
F3		$[\mathbf{d}'_{nk}/Q_{yk}]$	\mathbf{d}
F4		(x, y) (s_x, s_y) β/Q 的余数	
F5		β/Q 的商	
F6	$\Delta DCJ, I/Q_s$	$\Delta DCJ, I/Q_s$	$\Delta DCJ, I/Q_s$

(B) 块图象压缩编码

图12

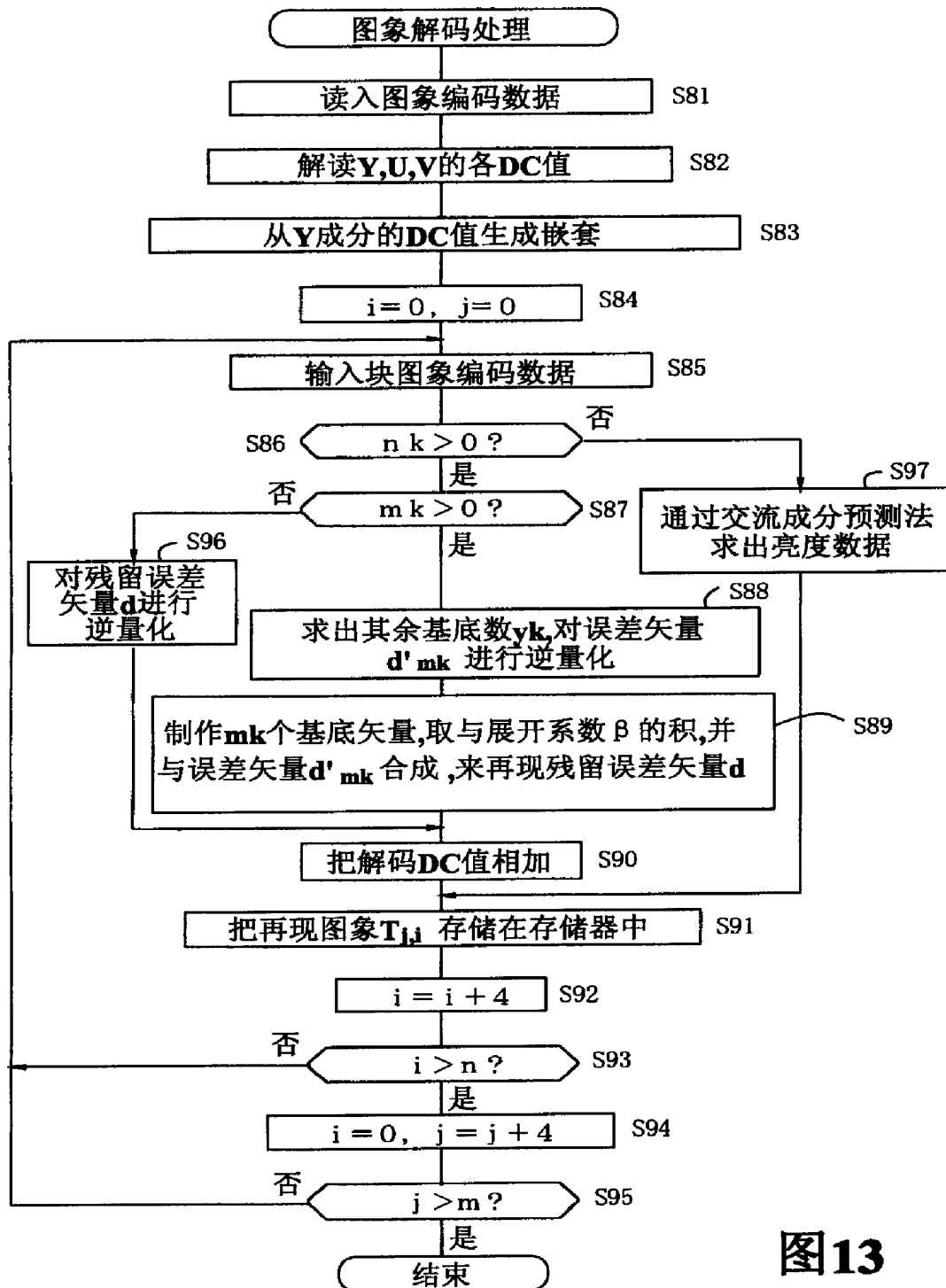
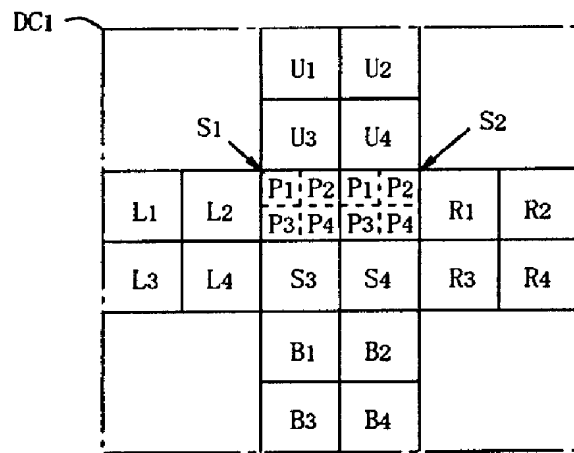
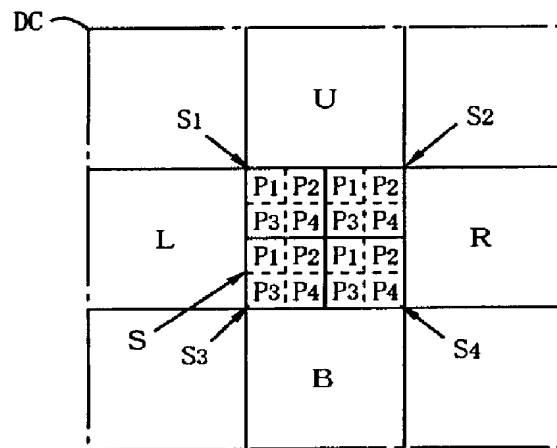


图13

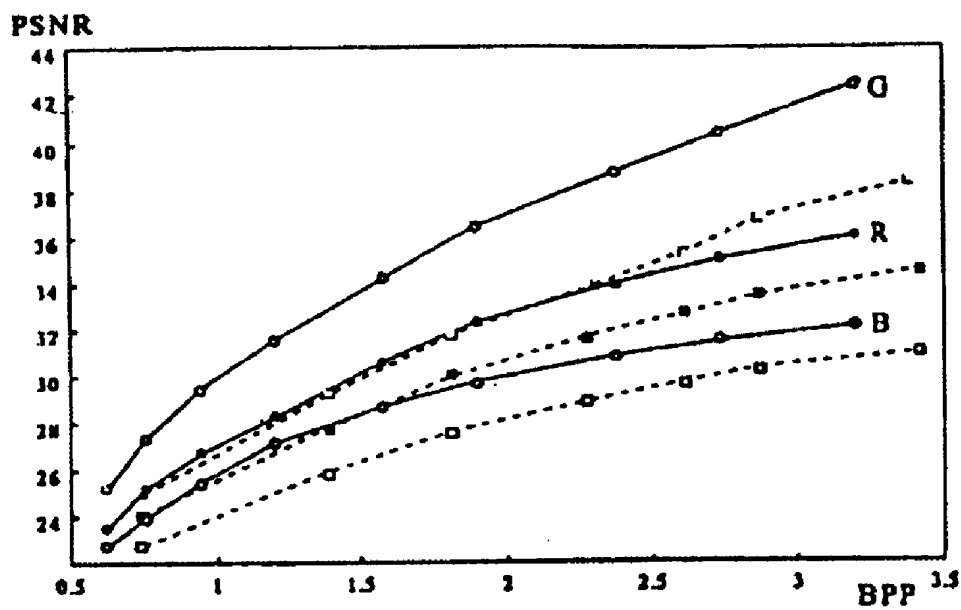


(A)

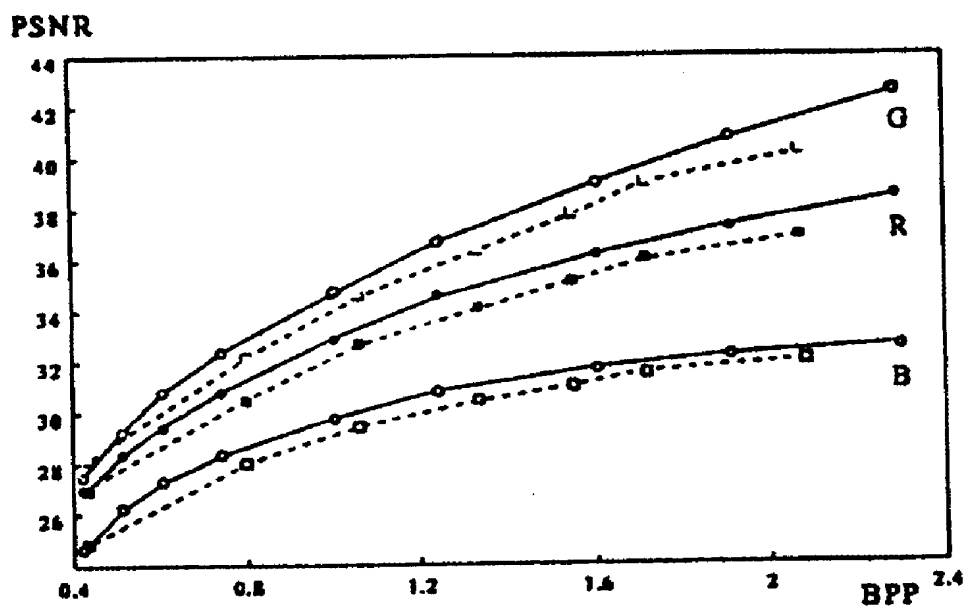


(B)

图14

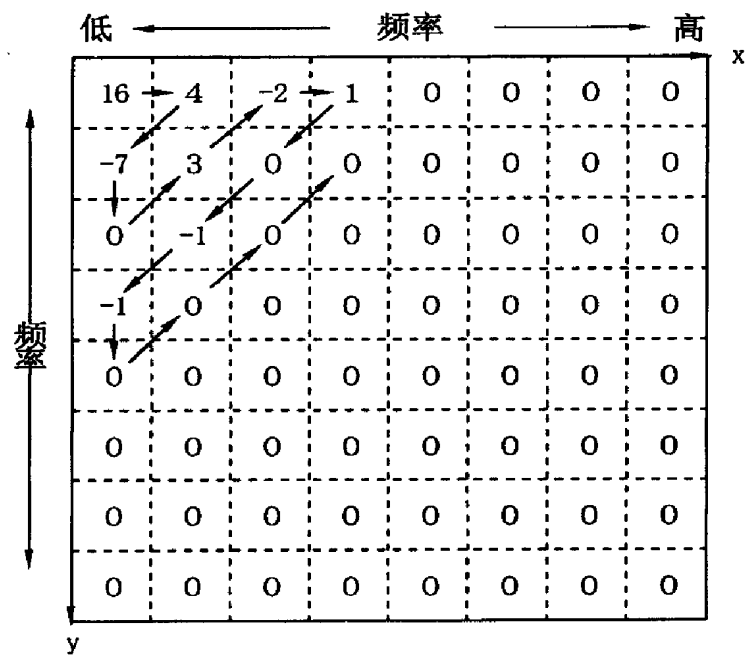


(A) 动画图象

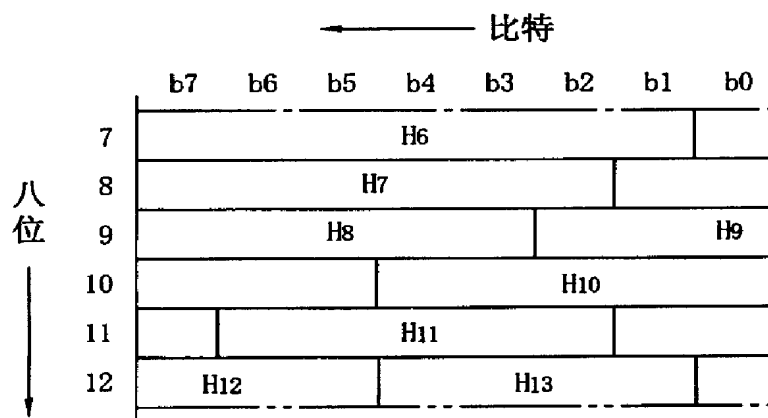


(B) CG图象

图15

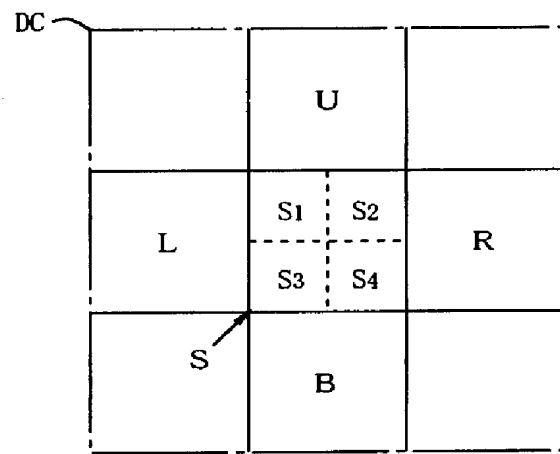


(A) DCT数据

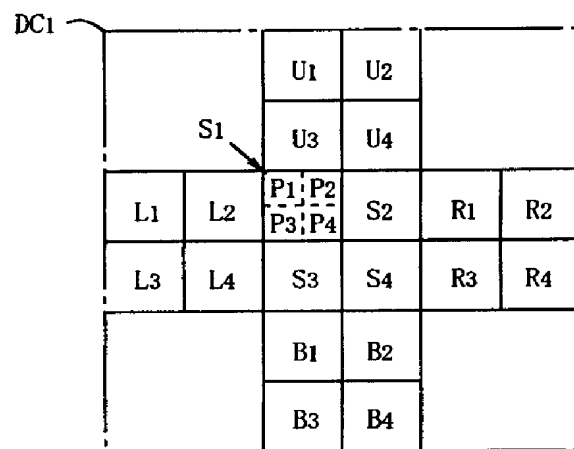


(B) 霍夫曼编码

图16



(A) 第1阶段



(B) 第2阶段

图17